



# Fast Models

Version 11.31

## Reference Guide

**Non-Confidential**

Copyright © 2017–2026 Arm Limited (or its affiliates).  
All rights reserved.

**Issue 00**

100964\_1131\_00\_en



## Fast Models Reference Guide

This document is Non-Confidential.

Copyright © 2017–2026 Arm Limited (or its affiliates). All rights reserved.

This document is protected by copyright and other intellectual property rights.

Arm only permits use of this document if you have reviewed and accepted [Arm's Proprietary Notice](#) found at the end of this document.

This document (100964\_1131\_00\_en) was issued on 2026-03-04. There might be a later issue at <https://developer.arm.com/documentation/100964>

The product version is 11.31.

See also: [Proprietary notice](#) | [Product and document information](#) | [Useful resources](#)

### Start reading

If you prefer, you can skip to [the start of the content](#).

### Intended audience

This document is written for software developers who are using the components and tools in the Fast Models portfolio to build custom platform models and integrate them with third-party tools and models.

### Inclusive language commitment

Arm values inclusive communities. Arm recognizes that we and our industry have used language that can be offensive. Arm strives to lead the industry and create change.

This document includes language that can be offensive. We will replace this language in a future issue of this document.

To report offensive language in this document, email [terms@arm.com](mailto:terms@arm.com).

### Feedback

Arm welcomes feedback on this product and its documentation. To provide feedback on the product, create a ticket on <https://support.developer.arm.com>.

To provide feedback on the document, fill the following survey: <https://developer.arm.com/documentation-feedback-survey>.

# Contents

<b>1. About the models.....</b>	<b>23</b>
1.1 Model capabilities.....	23
1.2 Running Arm Software Test Libraries (STL) on Fast Models.....	23
1.3 Quality level definitions.....	24
1.4 Fast Models accuracy.....	24
1.4.1 Accuracy of Code Translation model components.....	25
1.4.2 Timing accuracy of Fast Models.....	26
1.4.3 Bus traffic in Fast Models.....	27
1.4.4 Instruction prefetch in Fast Models.....	30
1.4.5 Out-of-order execution and write-buffers in Fast Models.....	30
1.4.6 Caches in Fast Models.....	31
1.4.7 Global exclusive monitor in Fast Models.....	31
1.5 Processor implementation.....	32
1.5.1 Caches in PV models.....	32
1.5.2 GICv3 in PV models.....	34
1.5.3 GICv4 in PV models.....	34
1.5.4 CP14 Debug coprocessor.....	35
1.5.5 TLBs in PV models.....	35
1.5.6 Memory access in PV models.....	35
1.5.7 Timing in PV models.....	36
1.5.8 Iris interactions with processor reset behavior.....	37
1.6 syncLevel definitions.....	38
1.7 Controlling and observing the syncLevel.....	39
1.8 User mode networking.....	41
1.9 TAP/TUN networking.....	42
1.9.1 TAP/TUN networking limitations.....	42
1.9.2 Setting up a network connection for Red Hat Enterprise Linux.....	42
1.9.3 Setting up a network connection for Ubuntu Linux.....	44
1.9.4 Configuring the networking environment for Linux.....	46
1.9.5 Solutions to networking issues on Linux.....	47
1.9.6 Disabling and re-enabling networking for Linux.....	47
1.9.7 Uninstalling networking for Linux.....	48

1.10 Using parameters to set port values.....	48
1.11 PVBUS C++ transaction and Tx_Result classes.....	49
1.11.1 Class pv::TransactionGenerator.....	49
1.11.2 TransactionGenerator efficiency considerations.....	50
1.11.3 Enum pv::AccessWidth.....	50
1.11.4 Class pv::Transaction.....	50
1.11.5 Class pv::ReadTransaction.....	51
1.11.6 Class pv::WriteTransaction.....	51
1.12 Visualisation library.....	52
1.12.1 LISA visualisation models.....	53
1.12.2 Visualisation library C++ classes.....	53
1.13 Timing annotation.....	58
1.13.1 CPI files.....	58
1.13.2 Instruction and data prefetching.....	66
1.13.3 Cache and TLB latency.....	68
<b>2. Protocols.....</b>	<b>70</b>
2.1 AMBAPV protocol.....	70
2.2 AMBAPVACE protocol.....	73
2.3 AMBAPVSignal protocol.....	76
2.4 AMBAPVSignalState protocol.....	77
2.5 AMBAPVValue protocol.....	77
2.6 AMBAPVValue64 protocol.....	78
2.7 AMBAPVValueState protocol.....	78
2.8 AMBAPVValueState64 protocol.....	79
2.9 AsyncSignalCallback protocol.....	79
2.10 AsyncSignalControl protocol.....	80
2.11 AudioControl protocol.....	80
2.12 CADIDisassemblerProtocol protocol.....	81
2.13 CADIProtocol protocol.....	82
2.14 CCI500_AddressDecoderProtocol protocol.....	86
2.15 CCIInterconnectControl protocol.....	87
2.16 ClockRateControl protocol.....	88
2.17 ClockSignal protocol.....	88
2.18 CompoundPortLisa protocol.....	90
2.19 CoprocBusProtocol protocol.....	90



2.20 CounterInterface protocol.....	94
2.21 DVMMMessage protocol.....	95
2.22 EventBus protocol.....	96
2.23 Feature protocol.....	96
2.24 FlashLoaderPort protocol.....	96
2.25 FrameTracingProtocol protocol.....	97
2.26 GICv3Comms protocol.....	97
2.27 GUIPollCallback protocol.....	98
2.28 ICS307Configuration protocol.....	99
2.29 InstructionCount protocol.....	99
2.30 KeyboardStatus protocol.....	100
2.31 LCD protocol.....	100
2.32 LCDLayoutInfo protocol.....	101
2.33 MMC_Protocol protocol.....	102
2.34 MMU_400_BASE_IDENTIFY protocol.....	104
2.35 MMU_400_Internals protocol.....	104
2.36 MMU_500_BASE_IDENTIFY protocol.....	104
2.37 MMU_500_Internals protocol.....	105
2.38 MouseStatus protocol.....	105
2.39 PASSwitchControl protocol.....	106
2.40 PCIDevice2ClientProtocol protocol.....	108
2.41 PCIeATC_get_if protocol.....	109
2.42 PChannel protocol.....	109
2.43 PL080_DMAC_DmaPortProtocol protocol.....	111
2.44 PL330_DMAC_DmaPortProtocol protocol.....	112
2.45 PMUEvent protocol.....	112
2.46 PS2Data protocol.....	112
2.47 PVBus protocol.....	113
2.48 PVBus2PCI2PCIDeviceProtocol protocol.....	116
2.49 PVBusBridgeControl protocol.....	118
2.50 PVBusCacheControl protocol.....	118
2.51 PVBusCacheDevice protocol.....	120
2.52 PVBusMapperControl protocol.....	121
2.53 PVBusOverTLMControl protocol.....	125
2.54 PVBusRouterControl protocol.....	125
2.55 PVBusSlaveControl protocol.....	126

2.56 PVC2C protocol.....	131
2.57 PVCacheDebugRam protocol.....	131
2.58 PVCacheMaintenance protocol.....	133
2.59 PVDevice protocol.....	138
2.60 PVTransactionMaster protocol.....	140
2.61 PVWriteBuffer_BarrierPort protocol.....	141
2.62 PVWriteBuffer_SErrorPort protocol.....	141
2.63 PVWriteBuffer_VmidBarrierPort protocol.....	142
2.64 SC_ClockRateControl protocol.....	142
2.65 SC_ClockSignal protocol.....	143
2.66 SC_VirtualEthernet protocol.....	145
2.67 SMMUv3AEMIdentifyProtocol protocol.....	145
2.68 SchedulerInterfaceControl protocol.....	146
2.69 SchedulerThreadControl protocol.....	147
2.70 SchedulerThreadEventControl protocol.....	148
2.71 SerialData protocol.....	148
2.72 Signal protocol.....	150
2.73 StateSignal protocol.....	150
2.74 SystemCCoprocBusProtocol protocol.....	151
2.75 SystemCPChannel protocol.....	151
2.76 SystemCoherencyInterface protocol.....	152
2.77 TZFilterControl protocol.....	152
2.78 TZSwitchControl protocol.....	153
2.79 TimerCallback protocol.....	154
2.80 TimerCallback64 protocol.....	155
2.81 TimerControl protocol.....	155
2.82 TimerControl64 protocol.....	156
2.83 VECBProtocol protocol.....	157
2.84 VGICComponentTraceExport protocol.....	157
2.85 VGICReportingProtocol protocol.....	158
2.86 Value protocol.....	158
2.87 ValueState protocol.....	159
2.88 ValueState_64 protocol.....	159
2.89 Value_64 protocol.....	160
2.90 VirtualEthernet protocol.....	160
2.91 VisEventRecorderProtocol protocol.....	160

2.92 v7_VGIC_Configuration_Protocol protocol.....	162
2.93 v8EmbeddedCrossTrigger_controlprotocol protocol.....	162
<b>3. Components.....</b>	<b>164</b>
3.1 Component differences.....	164
3.2 AEMv8RMPCT.....	167
3.3 AEMvACT.....	309
3.4 AEMvA_DSUCT.....	518
3.5 AHCI_SATA.....	732
3.6 AMBAPV2PVBus.....	733
3.7 AMBAPV2PVBusx4.....	735
3.8 AMBAPV2PVBusx8.....	736
3.9 AMBAPVACE2PVBus.....	736
3.10 AMBAPVSignal2SGSignal.....	738
3.11 AMBAPVSignal2SGSignalx1024.....	738
3.12 AMBAPVSignal2SGSignalx16.....	739
3.13 AMBAPVSignal2SGSignalx224.....	740
3.14 AMBAPVSignal2SGSignalx4.....	740
3.15 AMBAPVSignal2SGSignalx48.....	741
3.16 AMBAPVSignal2SGSignalx8.....	741
3.17 AMBAPVSignal2SGSignalx960.....	742
3.18 AMBAPVSignal2SGSignalx988.....	742
3.19 AMBAPVSignalState2SGStateSignal.....	743
3.20 AMBAPVSignalState2SGStateSignalx4.....	743
3.21 AMBAPVValue2SGValue.....	744
3.22 AMBAPVValue2SGValue64.....	745
3.23 AMBAPVValue2SGValue64x4.....	745
3.24 AMBAPVValue2SGValuex4.....	746
3.25 AMBAPVValue642SMMUv3AEMIdentify.....	746
3.26 AMBAPVValue642VECB.....	747
3.27 AMBAPVValueState2SGValueState.....	747
3.28 AMBAPVValueState2SGValueState64.....	748
3.29 AMBAPVValueState2SGValueState64x4.....	748
3.30 AMBAPVValueState2SGValueStatex4.....	749
3.31 ARMAEMv8MCT.....	749
3.32 ARMC1NanoCT.....	803

3.33 ARMC1NanoCT_C1ProCT.....	832
3.34 ARMC1NanoCT_C1ProCT_C1UltraCT.....	1002
3.35 ARMC1NanoCT_C1UltraCT.....	1234
3.36 ARMC1PremiumCT.....	1404
3.37 ARMC1ProCT.....	1431
3.38 ARMC1ProCT_C1UltraCT.....	1460
3.39 ARMC1UltraCT.....	1629
3.40 ARMCortexA320CT.....	1656
3.41 ARMCortexA32CT.....	1680
3.42 ARMCortexA34CT.....	1694
3.43 ARMCortexA35CT.....	1708
3.44 ARMCortexA510CT.....	1722
3.45 ARMCortexA510CT_CortexA710CT.....	1749
3.46 ARMCortexA510CT_CortexA710CT_CortexX2CT.....	1890
3.47 ARMCortexA510CT_CortexA710CT_CortexX3CT.....	2078
3.48 ARMCortexA510CT_CortexA715CT_CortexX3CT.....	2273
3.49 ARMCortexA520AECT.....	2474
3.50 ARMCortexA520CT.....	2502
3.51 ARMCortexA520CT_CortexA720CT.....	2530
3.52 ARMCortexA520CT_CortexA720CT_CortexX4CT.....	2700
3.53 ARMCortexA520CT_CortexA725CT.....	2932
3.54 ARMCortexA520CT_CortexA725CT_CortexX925CT.....	3102
3.55 ARMCortexA53CT.....	3333
3.56 ARMCortexA55CT.....	3349
3.57 ARMCortexA55CT_CortexA75CT.....	3367
3.58 ARMCortexA55CT_CortexA76CT.....	3447
3.59 ARMCortexA55CT_CortexA78CT.....	3526
3.60 ARMCortexA57CT.....	3606
3.61 ARMCortexA65AECT.....	3620
3.62 ARMCortexA65AECT_CortexA76AECT.....	3642
3.63 ARMCortexA65CT.....	3721
3.64 ARMCortexA710CT.....	3742
3.65 ARMCortexA715CT.....	3768
3.66 ARMCortexA720AECT.....	3796
3.67 ARMCortexA720CT.....	3824
3.68 ARMCortexA725CT.....	3851

3.69	ARMCortexA725CT_CortexX925CT.....	3877
3.70	ARMCortexA72CT.....	4046
3.71	ARMCortexA73CT.....	4059
3.72	ARMCortexA75CT.....	4073
3.73	ARMCortexA76AECT.....	4090
3.74	ARMCortexA76CT.....	4111
3.75	ARMCortexA77CT.....	4131
3.76	ARMCortexA78AECT.....	4151
3.77	ARMCortexA78CCT.....	4173
3.78	ARMCortexA78CT.....	4194
3.79	ARMCortexM0CT.....	4214
3.80	ARMCortexM0PlusCT.....	4219
3.81	ARMCortexM23CT.....	4225
3.82	ARMCortexM33CT.....	4234
3.83	ARMCortexM35PCT.....	4247
3.84	ARMCortexM3CT.....	4259
3.85	ARMCortexM4CT.....	4265
3.86	ARMCortexM52CT.....	4271
3.87	ARMCortexM55CT.....	4288
3.88	ARMCortexM7CT.....	4305
3.89	ARMCortexM85CT.....	4314
3.90	ARMCortexR4CT.....	4330
3.91	ARMCortexR52CT.....	4339
3.92	ARMCortexR52PlusCT.....	4354
3.93	ARMCortexR5x1CT.....	4369
3.94	ARMCortexR7x1CT.....	4379
3.95	ARMCortexR82AECT.....	4389
3.96	ARMCortexR82CT.....	4416
3.97	ARMCortexR8x1CT.....	4439
3.98	ARMCortexX1CCT.....	4451
3.99	ARMCortexX1CT.....	4472
3.100	ARMCortexX2CT.....	4492
3.101	ARMCortexX3CT.....	4516
3.102	ARMCortexX4CT.....	4545
3.103	ARMCortexX925CT.....	4571
3.104	ARMNeoverseE1CT.....	4597

3.105 ARMNeoverseN1CT.....	4618
3.106 ARMNeoverseN2CT.....	4638
3.107 ARMNeoverseN3CT.....	4663
3.108 ARMNeoverseV1CT.....	4687
3.109 ARMNeoverseV2CT.....	4709
3.110 ARMNeoverseV3AECT.....	4734
3.111 ARMNeoverseV3CT.....	4762
3.112 ARMSC000CT.....	4790
3.113 ARMSC300CT.....	4795
3.114 AddressTranslationUnit.....	4801
3.115 AndGate.....	4803
3.116 Ashbrook_SoC_SCC.....	4803
3.117 AsyncSignal.....	4807
3.118 AudioOut_File.....	4807
3.119 AudioOut_SDL.....	4808
3.120 BMU.....	4808
3.121 BP141_TZMA.....	4815
3.122 BP147_TZPC.....	4816
3.123 Base_PowerController.....	4817
3.124 BroadcastSignal2AMBAPVSignal.....	4823
3.125 CCI400.....	4824
3.126 CCI500.....	4828
3.127 CCI550.....	4836
3.128 CCN502.....	4844
3.129 CCN504.....	4847
3.130 CCN508.....	4850
3.131 CCN512.....	4854
3.132 CCSM_F1.....	4857
3.133 CFM.....	4864
3.134 CFMM.....	4865
3.135 CHBCR.....	4867
3.136 CI700.....	4868
3.137 CMM.....	4880
3.138 CMN600.....	4881
3.139 CMN600AE.....	4891
3.140 CMN600CMLHub.....	4903

3.141 CMN650.....	4904
3.142 CMN650R2.....	4915
3.143 CMN700.....	4926
3.144 CMN_S3.....	4943
3.145 CMSDK_Timer.....	4955
3.146 Clock2SystemC.....	4956
3.147 ClockDivider.....	4957
3.148 ClockGate.....	4958
3.149 ClockRateConversion.....	4959
3.150 ClockSelector.....	4960
3.151 ClockSignal2SC_ClockSignal.....	4964
3.152 ClockTimer.....	4964
3.153 ClockTimer64.....	4965
3.154 ClockTimerThread.....	4966
3.155 ClockTimerThread64.....	4967
3.156 Clock_Multiplexer.....	4967
3.157 ClusterClockControl.....	4968
3.158 Cluster_Temperature_Sensor.....	4974
3.159 CombinedMessagingUnit.....	4976
3.160 CombinedMessagingUnitAE.....	4993
3.161 CoprocBus2SystemC.....	5003
3.162 CounterInterface2SystemC.....	5004
3.163 D71.....	5005
3.164 DCSU.....	5012
3.165 DMA350.....	5013
3.166 DMC500.....	5019
3.167 DMC520.....	5021
3.168 DMC620.....	5023
3.169 DMC_400.....	5025
3.170 DMS_SUPER_CSR.....	5027
3.171 DTS.....	5028
3.172 DVFSM.....	5029
3.173 DebugAccessPort.....	5033
3.174 DebugROM.....	5034
3.175 DualClusterSystemConfigurationBlock.....	5045
3.176 DummyAPB.....	5048

3.177 ElfLoader.....	5050
3.178 EthosU55.....	5057
3.179 EthosU65.....	5059
3.180 EthosU85.....	5061
3.181 Firewall.....	5062
3.182 FlashLoader.....	5067
3.183 FrameTracingComponent.....	5069
3.184 FrequencyProbe.....	5072
3.185 GIC500.....	5073
3.186 GIC500_ClusterPorts.....	5099
3.187 GIC500_Filter.....	5125
3.188 GIC600.....	5152
3.189 GIC600AE.....	5166
3.190 GIC600AE_Filter.....	5180
3.191 GIC600_Filter.....	5195
3.192 GIC625.....	5209
3.193 GIC625_Filter.....	5215
3.194 GIC700.....	5220
3.195 GIC700_Filter.....	5245
3.196 GIC720AE.....	5270
3.197 GIC720AE_Filter.....	5294
3.198 GIC_400.....	5318
3.199 GIC_IRI.....	5325
3.200 GIC_IRI_Filter.....	5369
3.201 GICv3CommsLogger.....	5413
3.202 GICv3CommsPVBUS.....	5414
3.203 GICv3ProtocolChecker.....	5415
3.204 GICv5.....	5416
3.205 GUIPoll.....	5421
3.206 Generic_PLL.....	5422
3.207 HostBridge.....	5423
3.208 HostSerialInterface.....	5425
3.209 ICS307.....	5425
3.210 IDAU.....	5427
3.211 ILCU.....	5621
3.212 InstructionCount2SystemC.....	5623



3.213 InstructionCount2SystemCx4.....	5623
3.214 IntegrityChecker.....	5624
3.215 IntelStrataFlashJ3.....	5626
3.216 InterruptCombiner.....	5629
3.217 Interrupt_Router.....	5630
3.218 IoTSS3_ManagerSecurityController.....	5632
3.219 IoTSS3_MemoryProtectionController.....	5634
3.220 IoTSS3_SecureAccessConfig.....	5635
3.221 IoTSS3_SystemControl.....	5639
3.222 IoTSS_AccessControlGate.....	5642
3.223 IoTSS_PeripheralProtectionController.....	5643
3.224 Juno_sysregs.....	5644
3.225 KeyManagementUnit.....	5645
3.226 Kits2_Timer.....	5649
3.227 LCD2SystemC.....	5651
3.228 LS64TestingFIFO.....	5652
3.229 Labeller.....	5653
3.230 LabellerForDMA330.....	5654
3.231 LabellerForGPUProtMode.....	5656
3.232 LabellerIdauSecurity.....	5657
3.233 LabellerManagerIdExtendedIdUserFlag.....	5658
3.234 LabellerPAS.....	5659
3.235 LabellerUserSignals.....	5660
3.236 LifeCycleManager.....	5661
3.237 MHU320AE.....	5667
3.238 MMC.....	5671
3.239 MMU_400.....	5677
3.240 MMU_400_BASE.....	5700
3.241 MMU_500.....	5704
3.242 MMU_500_BASE.....	5727
3.243 MMU_600.....	5732
3.244 MMU_700.....	5753
3.245 MMU_720AE.....	5779
3.246 MMU_L1.....	5813
3.247 MMU_S3.....	5838
3.248 MSIRewriter.....	5872

3.249 Mali_C10.....	5875
3.250 Mali_C55.....	5876
3.251 Mali_C71.....	5879
3.252 Mali_C720AE.....	5881
3.253 Mali_C78.....	5882
3.254 Mali_Cxx_streaming_camera.....	5884
3.255 Mali_Cxx_streaming_sink.....	5885
3.256 Mali_G1.....	5887
3.257 Mali_G71.....	5890
3.258 Mali_G710.....	5892
3.259 Mali_G715.....	5895
3.260 Mali_G720.....	5898
3.261 Mali_G725.....	5901
3.262 Mali_G76.....	5903
3.263 Mali_G78AE.....	5906
3.264 Mali_T624.....	5908
3.265 MasterClock.....	5910
3.266 MemoryMappedCounterModule.....	5910
3.267 MemoryMappedGenericTimer.....	5913
3.268 MemoryMappedGenericWatchdog.....	5915
3.269 MessageHandlingUnit.....	5916
3.270 MessageHandlingUnitV2.....	5923
3.271 MessageHandlingUnitV3.....	5925
3.272 NI700.....	5930
3.273 NI710AE.....	5935
3.274 NOC_S3.....	5943
3.275 NonVolatileCounter.....	5950
3.276 OTPW.....	5952
3.277 OrGate.....	5953
3.278 PASSwitch.....	5954
3.279 PCIeATC.....	5956
3.280 PChannel2SystemC.....	5958
3.281 PL011_Uart.....	5958
3.282 PL022_SSP.....	5963
3.283 PL030_RTC.....	5964
3.284 PL031_RTC.....	5965

3.285	PL041_AACI.....	5966
3.286	PL050_KMI.....	5968
3.287	PL061_GPIO.....	5969
3.288	PL080_DMAC.....	5970
3.289	PL110_CLCD.....	5972
3.290	PL111_CLCD.....	5973
3.291	PL180_MCI.....	5975
3.292	PL192_VIC.....	5976
3.293	PL310_L2CC.....	5979
3.294	PL330_DMAC.....	5984
3.295	PL340_DMC.....	5994
3.296	PL350_SMC.....	5995
3.297	PL350_SMC_NAND_FLASH.....	6015
3.298	PL370_HDLCD.....	6017
3.299	PL390_GIC.....	6018
3.300	PLLClockControl.....	6027
3.301	PLLControl.....	6028
3.302	PMU.....	6029
3.303	PPUMTWakerequest.....	6032
3.304	PPUMultiThreadModeSwitch.....	6033
3.305	PPUv0.....	6034
3.306	PPUv1.....	6036
3.307	PPUv1_Cluster_Wakerequest_Logic.....	6043
3.308	PS2Keyboard.....	6044
3.309	PS2Mouse.....	6044
3.310	PVBus2AMBAPV.....	6045
3.311	PVBus2AMBAPVACE.....	6048
3.312	PVBus2AMBAPVx4.....	6051
3.313	PVBus2AMBAPVx8.....	6052
3.314	PVBus4KBTto1KBSplitter.....	6053
3.315	PVBusBridge.....	6054
3.316	PVBusCache.....	6055
3.317	PVBusDecoder.....	6056
3.318	PVBusExclusiveMonitor.....	6057
3.319	PVBusExclusiveSquasher.....	6060
3.320	PVBusGICv3Comms.....	6061

3.321 PVBUSLogger.....	6061
3.322 PVBUSMapper.....	6062
3.323 PVBUSMaster.....	6064
3.324 PVBUSModifier.....	6065
3.325 PVBUSModifierx2.....	6067
3.326 PVBUSRouter.....	6069
3.327 PVBUSSlave.....	6069
3.328 PVCohherentInterconnect.....	6072
3.329 PVMemoryProtectionEngine.....	6073
3.330 PVMetaDataController.....	6076
3.331 PVWriteBuffer.....	6080
3.332 PartialWriteDetector.....	6081
3.333 PchannelListener.....	6082
3.334 PowerStateGate.....	6083
3.335 RAMDevice.....	6084
3.336 RAM_ECC_Checker.....	6086
3.337 ROM.....	6087
3.338 RSE_CPU_Private_Region.....	6088
3.339 RSE_Integ_Regs.....	6089
3.340 RSE_SystemControl.....	6091
3.341 RandomNumberGenerator.....	6096
3.342 RealTimeLimiter.....	6097
3.343 RealtimeClockTimer.....	6098
3.344 RemapDecoder.....	6098
3.345 RootKeyStorage.....	6099
3.346 SC_ClockSignal2ClockSignal.....	6101
3.347 SDC600.....	6101
3.348 SGSignal2AMBAPVSignal.....	6103
3.349 SGSignal2AMBAPVSignalx16.....	6104
3.350 SGSignal2AMBAPVSignalx224.....	6105
3.351 SGSignal2AMBAPVSignalx256.....	6105
3.352 SGSignal2AMBAPVSignalx4.....	6106
3.353 SGSignal2AMBAPVSignalx48.....	6107
3.354 SGSignal2AMBAPVSignalx8.....	6107
3.355 SGSignalBuffer.....	6108
3.356 SGSignalBufferx16.....	6109

3.357	SGSignalBufferx2.....	6110
3.358	SGSignalBufferx4.....	6110
3.359	SGSignalBufferx8.....	6111
3.360	SGSignalBufferx988.....	6112
3.361	SGStateSignal2AMBAPVSignalState.....	6113
3.362	SGStateSignal2AMBAPVSignalStatex4.....	6113
3.363	SGValue2AMBAPVValue.....	6114
3.364	SGValue2AMBAPVValue64.....	6115
3.365	SGValue2AMBAPVValue64x4.....	6115
3.366	SGValue2AMBAPVValuex4.....	6116
3.367	SGValueState2AMBAPVValueState.....	6117
3.368	SGValueState2AMBAPVValueState64.....	6117
3.369	SGValueState2AMBAPVValueState64x4.....	6118
3.370	SGValueState2AMBAPVValueStatex4.....	6119
3.371	SI_L1.....	6119
3.372	SI_System_Ctrl_Regs.....	6129
3.373	SMCF.....	6131
3.374	SMMUv3AEM.....	6138
3.375	SMMUv3AEMIdentify2AMBAPVValue64.....	6200
3.376	SMMUv3TestEngine.....	6200
3.377	SMSC_91C111.....	6202
3.378	SP804_Timer.....	6204
3.379	SP805_Watchdog.....	6205
3.380	SP810_SysCtrl.....	6206
3.381	SSU.....	6209
3.382	STLBusGasket.....	6210
3.383	ScalableClockControl.....	6211
3.384	SchedulerInterface.....	6219
3.385	SchedulerThread.....	6219
3.386	SchedulerThreadEvent.....	6220
3.387	SecureAlarmManager.....	6220
3.388	SecurelCache.....	6223
3.389	SerialCrossover.....	6226
3.390	SignalDriver.....	6227
3.391	SignalInverter.....	6227
3.392	SignalLogger.....	6228

3.393 Signal_Multiplexer.....	6229
3.394 SimplePVBUSMaster.....	6229
3.395 SoC_SOR.....	6230
3.396 SoC_Temperature.....	6231
3.397 SwitchedClockControl.....	6232
3.398 SystemC2Clock.....	6238
3.399 SystemC2CoproccBus.....	6238
3.400 SystemC2CounterInterface.....	6239
3.401 SystemC2InstructionCount.....	6240
3.402 SystemC2InstructionCountx4.....	6240
3.403 SystemC2InstructionCountx8.....	6241
3.404 SystemC2LCD.....	6241
3.405 SystemC2PChannel.....	6242
3.406 SystemC2VirtualEthernet.....	6243
3.407 SystemC2v7VGICConfig.....	6243
3.408 SystemFMU.....	6244
3.409 SystemIdUnit.....	6245
3.410 System_RAS_Agent.....	6247
3.411 TRNG.....	6248
3.412 TZC_400.....	6249
3.413 TZFilterUnit.....	6260
3.414 TZIC.....	6260
3.415 TZSwitch.....	6261
3.416 TelnetTerminal.....	6263
3.417 Temperature.....	6265
3.418 TestbedGPIOConnector.....	6266
3.419 TrustedRAM.....	6266
3.420 UART_MUX.....	6268
3.421 Ufs.....	6268
3.422 UnusedPrimeCell.....	6270
3.423 V61.....	6270
3.424 V76.....	6274
3.425 VECB2AMBAPVValue64.....	6277
3.426 VHT_VIOBridge.....	6277
3.427 VHT_VSIBridge.....	6278
3.428 VHT_VSocket.....	6279

3.429 Value64Logger.....	6280
3.430 ValueLogger.....	6280
3.431 Value_Multiplexer.....	6281
3.432 VirtioBlockDevice.....	6282
3.433 VirtioBlockDeviceMMIO.....	6283
3.434 VirtioNetMMIO.....	6286
3.435 VirtioP9Device.....	6291
3.436 VirtioPCIBlockDevice.....	6294
3.437 VirtioRNG.....	6295
3.438 VirtualEthernet2SystemC.....	6298
3.439 VirtualEthernetCrossover.....	6299
3.440 VirtualEthernetHub3.....	6299
3.441 VisEventRecorder.....	6300
3.442 Visualisation_sdl2.....	6303
3.443 WarningMemory.....	6311
3.444 WideAndGate.....	6313
3.445 WideOrGate.....	6313
3.446 WideOrGate_12x4.....	6314
3.447 v7VGICConfig2SystemC.....	6315
3.448 v7_VGIC.....	6316
3.449 v8EmbeddedCrossTrigger_Matrix.....	6318
<b>4. Plug-ins for Fast Models.....</b>	<b>6320</b>
4.1 Loading a plug-in.....	6320
4.1.1 --plugin command-line option.....	6320
4.1.2 scx::scx_load_plugin() method.....	6320
4.1.3 FM_PLUGINS environment variable.....	6321
4.2 Customizing a plug-in.....	6322
4.3 ArchMsgTrace.....	6322
4.3.1 ArchMsgTrace - parameters.....	6324
4.4 ASTFplugin.....	6324
4.4.1 ASTFplugin - parameters.....	6325
4.4.2 ASTFplugin usage notes.....	6325
4.4.3 Additional ASTF support in Fast Models.....	6326
4.4.4 ASTF tools.....	6327
4.4.5 ASTFplugin FAQs.....	6327

4.5 CDE.....	6329
4.5.1 CDETester.....	6330
4.5.2 CDETester - parameters.....	6331
4.5.3 CDEConstant.....	6332
4.5.4 CDEConstant - parameters.....	6332
4.5.5 Implementing a CDE plug-in.....	6332
4.5.6 CDE API.....	6335
4.6 CDELoader.....	6342
4.6.1 CDELoader - parameters.....	6344
4.6.2 ACI library implementation.....	6344
4.6.3 ACIConstant example ACI library.....	6345
4.6.4 Custom instruction mnemonics.....	6346
4.6.5 ACI library API.....	6346
4.7 Crypto.....	6361
4.7.1 Crypto - parameters.....	6363
4.8 GenericCounter.....	6364
4.8.1 GenericCounter - parameters.....	6365
4.9 GenericTrace.....	6365
4.9.1 GenericTrace - parameters.....	6366
4.9.2 GenericTrace plug-in usage example.....	6367
4.9.3 Mapping between SYSREG_UPDATE trace sources and SPSR registers.....	6369
4.10 ListTraceSources.....	6370
4.10.1 ListTraceSources - parameters.....	6370
4.11 RunTimeParameterTest.....	6371
4.12 Sidechannel.....	6371
4.12.1 Sidechannel - parameters.....	6371
4.13 TarmacText.....	6371
4.14 TarmacTrace.....	6372
4.14.1 TarmacTrace - parameters.....	6373
4.14.2 TarmacTrace file format.....	6375
4.14.3 Tarmac Trace output example.....	6377
4.14.4 Branch Target Exception event trace.....	6379
4.14.5 Cache maintenance trace.....	6380
4.14.6 Cache content trace.....	6381
4.14.7 ETE write trace.....	6382
4.14.8 Event trace.....	6382



4.14.9 Granule protection table walk trace.....	6385
4.14.10 Instruction trace.....	6387
4.14.11 Memory bus trace.....	6389
4.14.12 POE2 fill and evict trace.....	6391
4.14.13 POE2 table fetch trace.....	6392
4.14.14 Processor memory access trace.....	6394
4.14.15 Processor memory update trace.....	6398
4.14.16 Program flow trace.....	6400
4.14.17 Register trace.....	6401
4.14.18 SIGNAL trace.....	6402
4.14.19 SPE event trace.....	6403
4.14.20 TLB trace.....	6403
4.14.21 Translation table walk trace.....	6405
4.14.22 vMTE tag VA access trace.....	6407
4.15 ToggleMTIPlugin.....	6408
4.15.1 ToggleMTIPlugin - parameters.....	6408
4.15.2 How to use ToggleMTIPlugin.....	6409
<b>5. SystemC Export.....</b>	<b>6410</b>
5.1 About SystemC Export.....	6410
5.2 Bridging between LISA+ and SystemC.....	6410
5.3 SystemC Export generated ports.....	6412
5.4 SystemC Export limitations.....	6413
5.5 scx API.....	6414
5.6 scx_simcallback_if class.....	6438
5.7 scx_simcontrol_if class.....	6439
<b>6. Scheduler API.....</b>	<b>6444</b>
6.1 Intended mapping of the Scheduler API onto SystemC/TLM.....	6444
6.2 Accessing SchedulerInterfaceForComponents from a modeling component.....	6445
6.3 FrequencyObserver class.....	6446
6.4 FrequencySource class.....	6447
6.5 GlobalFrequencySource class.....	6449
6.6 SchedulerCallback class.....	6450
6.7 SchedulerInterfaceForComponents class.....	6451
6.8 SchedulerObject class.....	6463
6.9 SchedulerRunnable class.....	6464

6.10 SchedulerRunnableWithGetRunnableName class.....	6467
6.11 SchedulerThread class.....	6467
6.12 Tag class.....	6469
6.13 ThreadSignal class.....	6469
6.14 Timer class.....	6470
6.15 TimerCallback class.....	6472
<b>7. FastRAM.....</b>	<b>6473</b>
7.1 How to enable FastRAM.....	6473
7.2 FastRAM configuration file syntax.....	6474
7.3 FastRAM configuration file example.....	6475
7.4 FastRAM limitations.....	6476
<b>8. Fast Models examples.....</b>	<b>6478</b>
8.1 LISA examples.....	6478
8.2 Build and run an FVP example.....	6479
8.3 LISAPlus examples.....	6480
8.4 MTI examples.....	6481
8.5 SystemCExport examples.....	6482
8.6 Build and run an EVS platform example.....	6482
8.7 Build and run an SVP example.....	6483
<b>Proprietary notice.....</b>	<b>6485</b>
<b>Product and document information.....</b>	<b>6487</b>
Product status.....	6487
Revision history.....	6487
Conventions.....	6489
<b>Useful resources.....</b>	<b>6492</b>

# 1. About the models

Programmers View (PV) models of processors and devices work at a level where functional behavior is equivalent to what a programmer would see using the hardware.

They sacrifice timing accuracy to achieve fast simulation execution speeds. You can use the PV models for confirming software functionality, but you must not rely on the accuracy of cycle counts, low-level component interactions, or other hardware-specific behavior.

## 1.1 Model capabilities

Fast Models attempt to accurately model the hardware, but compromises exist between speed of execution, accuracy, and other criteria. A processor model might not match the hardware under certain conditions.

Fast Models can:

- Accurately model instructions.
- Correctly execute architecturally-correct code.
- Model some unpredictable behavior.

Fast Models cannot:

- Validate the hardware.
- Model all unpredictable behavior.
- Model cycle counting.
- Model timing-sensitive behavior.
- Model SMP instruction scheduling.
- Measure software performance.
- Model bus traffic.

### Related information

[Caches in Fast Models](#) on page 30

## 1.2 Running Arm Software Test Libraries (STL) on Fast Models

Fast Models does not support running the entire STL.

The reasons are:

- Some micro-architectural features implemented in the RTL are not implemented in the Fast Model and therefore the model's behavior might deviate from the RTL.
- Some peripheral support must be included to enable parts of the STL. For example SBIST controller support is available in the RTL but not in the Fast Model.

Fast Models can be used for integration of the STL into your software stack, but running the STL might not perform accurate measurement and validation.

## Related information

[Arm Software Test Libraries](#)

## 1.3 Quality level definitions

The documentation for each model of Arm® IP includes one or more quality level statements to indicate how complete the model's implementation is for each supported revision of the IP.

**Table 1-1: Quality level definitions**

Quality level	Definition
Alpha support	The model implementation is at an early stage and is likely to change in future releases. There might be significant defects or limitations in the model.
Preliminary support	The model implementation is complete or almost complete. Testing is nearly to the level of a fully-supported model. However, changes might still occur and there are likely to be known defects present.
Full support	Also called Release quality. Support is complete and the model has been tested as fully as possible. The modeled IP has reached its LAC milestone.

## 1.4 Fast Models accuracy

Fast Models aim to be accurate from the point of view of the program running on the processors. This section describes areas where Fast Models differ from hardware.

Software is able to detect differences between hardware and the model, but these differences generally depend on behavior that is not precisely specified. For example, it is possible to detect differences in the exact timings of instructions and bus transactions, effects of speculative prefetch and cache victim selection.

Certain classes of behavior are specified as unpredictable and these cases are detectable by software. A program that relies on such behavior, even unintentionally, is not guaranteed to work reliably on any device, or on a Fast Model. Programs that exploit this behavior might execute differently between the hardware and the model.

Fast Models do not attempt to accurately model bus transactions. PVBus provides the infrastructure to ensure that the program gets the correct results.

### 1.4.1 Accuracy of Code Translation model components

Code Translation (CT) components differ in some ways from the modeled IP.

See also the Limitations sections in the [Components](#) chapter for component-specific limitations.

- They do not model cycle timing. In aggregate, all instructions execute in one processor master clock cycle, except for Wait For Interrupt.
- Write buffers are not modeled on all processors.
- Most aspects of TLB behavior are implemented in the models. The TLB memory attribute settings are used when stateful cache is enabled.
- No device-accurate MicroTLB is implemented.
- Device-accurate modeling of multiple TLBs is off by default.
- A single memory access port is implemented. The port combines accesses for instruction, data, DMA, and peripherals. Configuration of the peripheral port memory map register is ignored.
- All memory accesses are atomic and are performed in Programmer's View order. Unaligned accesses are always performed as byte transfers.
- Some instruction sequences are executed atomically so that system time does not advance during their execution. This difference in behavior can affect sequential accesses of device registers where devices are expecting time to move on between each access.
- Interrupts are not taken at every instruction boundary.
- Integration and test registers are not implemented.
- Models cannot run Software Test Libraries (STLs).
- Not all CP14 debug registers are implemented on all processors.
- Breakpoint types that the models support directly are:
  - Single address unconditional instruction breakpoints.
  - Single address unconditional data breakpoints.
  - Unconditional instruction address range breakpoints.
- Pseudoregisters in the debugger support processor exception breakpoints. Setting an exception register to a nonzero value stops execution on entry to the associated exception vector.
- Cluster models do not simulate all cores at the same time. They execute a number of instructions on each core in turn. There can be a bias in the order in which cores run after a restart (for example, core 0 always runs first), so the simulation might hit breakpoints on the favored core more often.
- Performance counters are not implemented on all models.
- Some models implement caches, although all processor models implement cache control registers.
- ECC and parity schemes are hardware-specific so are not modeled.
- Models use a simplified view of the external buses.
- Clusters based on DSU-110 and later support these modes:

- OFF
- ON
- DBG\_RECOV

## 1.4.2 Timing accuracy of Fast Models

Fast Models do not model instruction timing accurately. The simulation as a whole has a very accurate concept of timing, but the Code Translation (CT) processors do not claim to dispatch instructions with device-like timing.

In general, a processor issues a set of instructions, called a quantum, at the same point in simulation time, and then waits for some amount of time before executing the next quantum.

Timing is arranged so that on average the processor executes one instruction per clock cycle, although [Timing Annotation](#) can cause the cycle count and instruction count to differ.

The consequences of this are:

- The perceived performance of software running on the model differs from real-world software. In particular, memory accesses and arithmetic operations all take a significant amount of time.
- A program might be able to detect the quantized execution behavior of a processor, for example by polling a high-resolution timer.
- All instructions in a quantum are effectively atomic.



Note

This might mask some race-condition bugs in software.

---

The following conditions can cause the processor to yield to another thread before a quantum expires:

- Non-DMI memory transactions
- Signal changes
- WFE and WFI instructions
- Barrier instructions
- Yield instructions
- Generic Timer accesses

### 1.4.3 Bus traffic in Fast Models

PVBus is the bus protocol that is used to model all memory-like buses in Fast Models.

#### PVBus components

PVBus is an internal protocol. The interface to it is through the PVBus components, which abstract the internal details. These are components with the LISA property `component_type = "Bus"`. They provide functionally-accurate communication between bus masters and slaves. They are not software implementations of specific hardware, but instead are abstract components that are required by the software model environment.

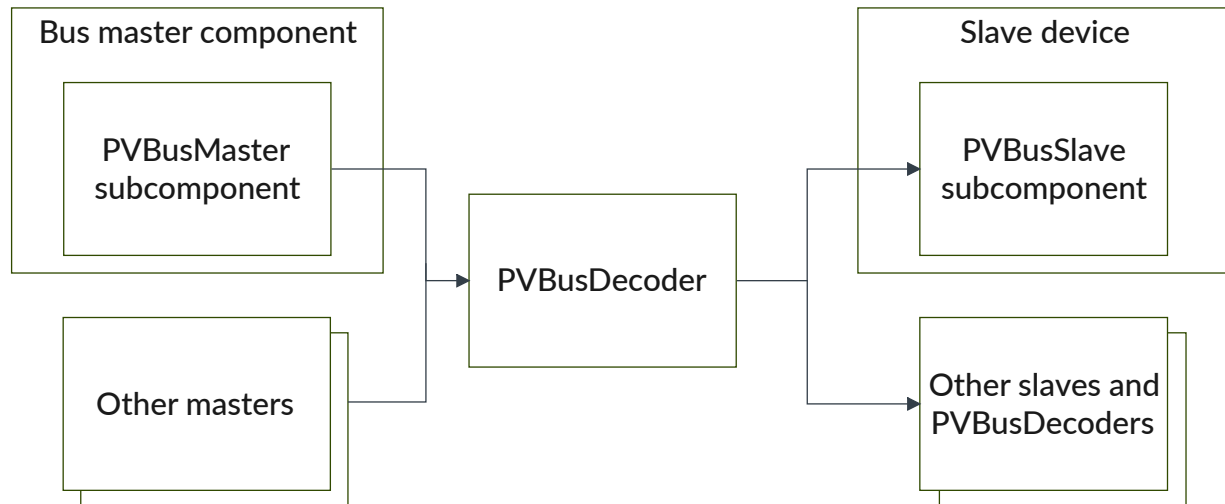
Each bus master must contain a [PVBusMaster](#) subcomponent, and each bus slave must contain a [PVBusSlave](#) subcomponent. These subcomponents provide PVBus master and slave ports. Each PVBus master port can only connect to one slave, but any number of other masters can connect to the same slave.

[PVBusDecoder](#) components can be added to the bus system. Each of these permits its masters to connect to multiple slaves, each associated with a different bus address range.

PVBusSlave subcomponents provide built-in support for declaring address ranges that are one of the following:

- Memory-like
- Device-like
- Abort
- Ignore

All communication over the PVBus is performed using transactions that are generated by PVBusMaster subcomponents and fulfilled by PVBusSlave components. Transactions have a 32-bit Master ID, which is the ID of the bus master. Transactions can be routed to the slave device through its PVBusSlave subcomponent. When configured, the PVBusSlave can handle memory-like transactions efficiently without having to route these transactions to the slave device. Transactions are atomic unless slave devices block transactions, for example an SMMU with stall mode enabled. A slave device that can block transactions and all its upstream bus components must be re-entrant safe for bus transactions.

**Figure 1-1: Sample bus topology**

Fast Models provides some example PVBus systems:

**\$PVLIB\_HOME/examples/LISA/Common/LISA/RemapDecoder.lisa**

This example dynamically modifies routing of requests based on a remap signal, using the [TZSwitch](#) component.

**\$PVLIB\_HOME/examples/LISA/BusComponents/**

This directory contains example components that show various ways of using the PVBus interface.

## PVBus optimizations

PVBus can simulate the behavior of individual bus transactions passing through a hierarchy of bus fabric, but it uses the following optimizations:

- PVBus generally decodes the path between a bus master and bus slave the first time a transaction is issued. All subsequent transactions to the same address are automatically sent to the same slave, without passing through the intervening fabric.
- For accesses to normal memory, the master can cache a pointer to the host storage that holds the data contents of the memory. The master can read and write directly to this memory without generating bus transactions.
- For instruction-fetch, and for operations such as repeated DMA from framebuffer memory, PVBus provides an optimization called snooping, which informs the master if anyone else could have modified the contents of memory. If no changes have occurred, the master can avoid the need to re-read memory contents.

If a piece of bus fabric wants to intercept and log all bus transactions, it can defeat these optimizations by claiming to be a slave device. It can then log all transactions and can reissue identical transactions on its own master port. However, doing this slows all bus transactions and significantly impacts simulation performance.





If direct accesses to memory by the CT engine are intercepted by the fabric, the processor is forced to single step. Execution is much slower than normal operation with translated code.

---

## Differences between modeled bus traffic and real traffic

The bus traffic generated by Fast Models is not representative of real traffic:

### Timing differences

Reordering and buffering of memory accesses, out-of-order execution, speculative prefetch, and drain-buffers can cause timing differences. They are not modeled, since they are not visible to the programmer except in situations where a cluster program contains race conditions that violate serial-consistency expectations.

### Bus contention

Fast Models do not model the time taken for a bus transaction, so they cannot model the effects of multiple transactions contending for bus availability.

### Size of access

Fast Models do not attempt to generate the same types of burst transaction from the processor for accesses to multiple consecutive locations. PVBUS only supports burst transactions of type INCR.

### Instruction fetch

The behavior of the instruction prefetch unit of a processor is not modeled to match the hardware implementation.

### Behavioral differences

In some software, the trace of instruction execution depends on timing effects. For example, if a loop polls a device waiting for a 10ms time-out, the number of iterations of the polling loop depends on the rate of instruction execution.

### Other differences:

- PVBUS does not support any type of write strobes.
- PVBUS does not support Quality of Service (QoS) or region values.
- Transactions cannot cross a 4KB boundary.
- Barriers and DVM messages are not transmitted on the PVBUS.

## Related information

[Instruction prefetch in Fast Models](#) on page 29

### 1.4.4 Instruction prefetch in Fast Models

The CT engine in the processor models relies on Fast Models PVBUS optimizations. It only performs code-translation if it has been able to prefetch and snoop the underlying memory. It then need not issue bus transactions until the snoop handling detects an external modification to memory.

If the CT engine cannot get prefetch access to memory, it drops to single-stepping. This single-stepping is very slow (~1000x slower than translated code execution).

Real processors attempt to prefetch instructions ahead of execution and predict branch destinations to keep the prefetch queue full. The instruction prefetch behavior of a processor can be observed by a program that writes into its own prefetch queue (without using explicit barriers). The architecture does not define the results.

The CT engine processes code in blocks. The effect is as if the processor filled its prefetch queue with a block of instructions, then executed the block to completion. As a result, this virtual prefetch queue is sometimes larger and sometimes smaller than the corresponding hardware. In the current implementation, the virtual prefetch queue can follow small forward branches.

With an L1 instruction cache turned on, the instruction blocksize is limited to a single cache-line. The processor ensures that a line is present in the cache at the point where it starts executing instructions from that line.

In real hardware, the effects of the instruction prefetch queue are to cause additional fetch transactions. Some of these are redundant because of incorrect branch prediction. This causes extra cache and bus pressure.

### 1.4.5 Out-of-order execution and write-buffers in Fast Models

Hardware memory is Weakly Ordered, but Fast Models memory is Strongly Ordered.

The CT implementation executes instructions sequentially. One instruction is retired before the next starts to execute. In a real processor, multiple memory accesses can be outstanding, and can complete in a different order from their program order. Writes can also be delayed in a write-buffer.

The programmer-visible effects of these behaviors are defined in the architecture as the Weakly Ordered memory model, which the programmer must be aware of when writing lock-free cluster code.

Within Fast Models, memory accesses happen in program order, effectively as if all memory is Strongly Ordered.

## 1.4.6 Caches in Fast Models

Fast Models with cache-state modeling enabled can replicate some, but not all, types of failure-case.

The effects of caches are programmer-visible because they can cause a single memory location to exist as multiple inconsistent copies. If caches are not correctly maintained, reads can observe stale copies of locations, and flushes/cleans can cause writes to be lost.

There are three ways in which incorrect cache maintenance can be programmer-visible:

### **From the D-side interface of a single processor**

The only way of detecting the presence of caches is to create aliases in the memory map, so that the same range of physical addresses can be observed as both cached and non-cached memory.

### **From the D-side of a single processor to its I-side**

Stale instruction data can be fetched when new instructions have been written by the D-side. This can either be because of deliberate self-modifying code, or as a consequence of incorrect OS demand paging.

### **Between one processor and another device**

For example, another processor in a non-coherent MP system, or an external DMA device.

Fast Models with cache-state modeling enabled can replicate all of these failure cases. However, they do not attempt to reproduce the following effects of caches:

- Changes to timing behavior of a program because of cache hits/misses (because the timing of memory accesses is not modeled).
- Ordering of line-fill and eviction operations.
- Cache usage statistics (because the models do not generate accurate bus traffic).
- Device-accurate allocation of cache victim lines (which is not possible without accurate bus traffic modeling).
- Write-streaming behavior where a cache spots patterns of writes to consecutive addresses and automatically turns off the write-allocation policy.

It is not possible to insert devices between the processor and its L1 caches. In particular, you cannot model L1 traffic, although you can tell the model not to model the state of L1 caches.

## 1.4.7 Global exclusive monitor in Fast Models

A monitor for cacheable nonshared and shared memory is always implemented, but the implementation differs depending on whether `cache-state-modelled` is `true` or `false`.

- When `cache-state-modelled` is `true`, caches are modeled, so exclusiveness is handled by the cache coherency protocol. Each line in a coherent cache records whether it is exclusive. When another core writes to an exclusive line, it is invalidated in other caches.

- When `cache-state-modelled` is `false`, the cache state is not modeled, so there are no cache lines or coherency. To provide the same functionality from a software perspective, an exclusive monitor is instantiated internally to maintain coherency.

The RAMDevice component and PVBus to AMBAPV bridges do not contain global monitors. As a result, exclusive stores that reach a RAMDevice fail unless they go through an exclusive monitor. Some platforms might need a global monitor outside of the cache coherency domains. These platforms must include a system-level monitor in the same place in the bus hierarchy as in the hardware. See the FVP\_VE and FVP\_Base example platforms for examples of how to use the PVBusExclusiveMonitor component.

## 1.5 Processor implementation

This section outlines the differences between the code translation models and the hardware.

### 1.5.1 Caches in PV models

Some processor models have PV-accurate caches, but others do not model Level 1 or Level 2 caches.

Cores that have PV-accurate cache implementation provide a functionally-accurate model. For more information, see the processor component descriptions.

Other PV models do not model Level 1 or Level 2 caches. The system coprocessor registers related to cache operations permit cache-aware software to work, but in most cases they only check register access permissions.

The registers affected on all code translation processor models are:

- Invalidate and/or Clean Entire ICache/DCache.
- Invalidate and/or Clean ICache/DCache by MVA.
- Invalidate and/or Clean ICache/DCache by Index.
- Invalidate and/or Clean Both Caches.
- Cache Dirty Status.
- Data Write Barrier.
- Data Memory Barrier.
- Prefetch ICache Line.
- ICache/DCache lockdown.
- ICache/DCache master valid.

### 1.5.1.1 Functional caches in Fast Models

Fast Models implement two types of cache model:

#### Register model

Provides all the cache control registers so that cache operations succeed, but does not model the state of the cache. All accesses go to memory.

#### Functional model

Tracks cache state and its contents at each level of the memory hierarchy. Incorrect cache management might return incorrect data, as it would on real hardware.

Fast Models provide:

- System IPs that support caches.
- Register models of caches on all processors that support caches and also the PL310 cache controller (PL310\_L2CC).
- Functional models of caches integrated into Cortex cores.



Note

For cache-state modeling to work, all components within the coherency domain must have consistent cache-state modeling settings. Support for cache state modeling beyond the CPUs is deprecated and will be removed in a future release. When support is removed from the interconnects, this means that cache state modeling can only be enabled if the coherency boundary is before the interconnect, for example at the cluster boundary.

For an Arm®v8-A core with an L2 cache, the configuration parameters are:

#### **icache-state\_modelled**

Set whether L1 I-cache has stateful implementation. Instructions at L2 or L3 are not cached in Fast Models.

#### **dcache-state\_modelled**

Set whether D-cache has stateful implementation at L1, L2, and L3.

### 1.5.1.2 Performance impact of functional caches in Fast Models

Enabling functional cache modeling is likely to reduce performance.

Enable the L1 and L2 functional caches together. For consistent system operation, Arm recommends that you either disable functional behavior completely or enable it for both I and D L1 caches and the L2 cache.

Cache enablement must be system wide. If you enable cache state modeling in any component then you must enable it in all components in the system, including all cores (L1 and L2) and any external cache controller (such as the PL310\_L2CC) and any interconnect that has caches.

If platform memory is being modeled outside of the Fast Models environment, for example in a SystemC environment, use of functional cache modeling might improve performance if there is no other fast route to memory.

## 1.5.2 GICv3 in PV models

The PV models implement the Generic Interrupt Controller architecture version 3 (GICv3).

The GICv3 architecture defines two parts:

- The core interface (integrated into the core)
- The Interrupt Redistribution Infrastructure (IRI)

You can configure all Arm®v8-A cores to include a GICv3 interface. You can integrate a separate GIC\_IRI component into your platforms. Communication between the core and the IRI is over an architected packet interface. An internal communication protocol represents the packets that pass over this interface.

You can configure the GICv3 models in some platforms to act as though they were GICv2 or GICv2-M models. Even in this mode, you need a GIC\_IRI component and a supported core. Configure them to comply with the same standard.

Models have the following limitations:

- Support for the GITS\_CTLR.Quiescent bit is not complete.
- Support for ITS save/restore is not complete. Configuration stays within the model and it does not use allocated memory.
- GICD\_CTLR.RWP does not perform adequately. This difference is only an issue if you use the distributor in systems with delaying interface between the distributor and the cores. Do not use this version of the model for simulation of the GIC in a setup where interfaces are not instantly reactive.

Set the environment variable `FASTSIM_GIC_MEMORY_MAP` to 1 to print to `stderr` the memory map of certain models that are included in the platform being run. This functionality is available for all GICv3 and later models.

## 1.5.3 GICv4 in PV models

GICv4 is an extension of the GICv3 architecture. It allows the direct injection of LPIs into a virtualized system through the `virtual-lpi-support` parameter of the `GIC_IRI` or `GIC_IRI_Filter` component.

In addition to requiring the presence of an ITS that is configured as shown in [GIC\\_IRI](#), GICv4 requires you to enable the virtual LPIs feature and to configure a virtual PE table using the parameters shown in this example:

```
"virtual-lpi-support"=true,
```

```
"GITS_BASER4-type"=2 //Type 2 is Virtual PEs.  
                      //Such a table is needed for GICv4 functionality.
```

## 1.5.4 CP14 Debug coprocessor

Some models fully implement the CP14 Debug coprocessor registers. Other models only implement the DSCR register. This register reads as 0 and ignores writes.

External debugging must be used to debug systems containing PV models.

## 1.5.5 TLBs in PV models

The PV models implement Translation Lookaside Buffers (TLBs) and model most aspects of TLB behavior.



Note

If the `device-accurate-tlb` parameter is set to `false`, the simulation uses a different number of TLBs if this improves simulation performance. The simulation is architecturally accurate, but not device accurate. Architectural accuracy is almost always sufficient. Set `device-accurate-tlb` to `true` if you require device accuracy.

These TLB registers do not have working implementations:

- Primary memory remap register.
- Normal memory remap register.

In addition, the simulation does not distinguish peripheral accesses from data accesses, so it ignores configuration of the peripheral port memory remap register.



Note

The models do not implement device-accurate MicroTLBs, or system coprocessor registers related to MicroTLB state.

## 1.5.6 Memory access in PV models

PV models use a PVBUSMaster subcomponent to communicate with slaves in a SimGen-generated system. This provides efficient access to memory-like slaves and relatively efficient access to device-like slaves.

Memory access in PV models differs from real hardware to enable fast modeling of the processor:

- All memory accesses are performed in programmer view order.
- Unaligned accesses, where permitted, are always performed as byte transfers.

In addition, some PV models do not use all the transaction states available in a PVBUS transaction. The Privileged and Instruction flags are set correctly for Arm®v7 processors but might not be set correctly in earlier architectures. However all memory accesses are atomic so `swp` instructions behave as expected.

### 1.5.6.1 I-side access in PV models

PV models cache translations of instructions fetched from memory-like slaves. The models might not perform further access to those slaves for significant periods. A slave can force the model to reread the memory by declaring that the memory has changed.

PV models do not model a prefetch queue but the code translation mechanism effectively acts as a prefetch queue of variable depth. Arm recommends that you follow the standards in the [Arm® architecture specifications](#) for dealing with prefetch issues, such as self modifying code, and use appropriate cache flushing and synchronization barriers.

Translation of instructions only occurs for memory-like slaves, which are those declared by devices as having type `p_v::ACCESSMODE_MEMORY`. Instructions fetched from device-like slaves are repeatedly fetched, decoded, and executed, significantly slowing down model performance.

### 1.5.6.2 D-side access in PV models

PV models cache references to the underlying memory of memory-like slaves, and might not perform further accesses to those slaves over the bus for significant periods.

Slaves declared as type `p_v::ACCESSMODE_MEMORY` provide the fastest possible memory access for PV processors.

Slaves declared as type `p_v::ACCESSMODE_DEVICE` are normally used for peripheral access.

## 1.5.7 Timing in PV models

Programmers View (PV) models are loosely timed.

- Caches and write buffers are not modeled, so all memory access timing is effectively zero wait state.
- All instructions execute, in aggregate, in one cycle of the component master clock input.
- Interrupts are not taken at every instruction boundary.
- Some sequences of instructions are executed atomically, ahead of the master clock of a component, so that system time does not advance during execution. This difference in behavior can affect sequential access of device registers, where devices are expecting time to move on between accesses.
- DMA to and from Tightly Coupled Memory (TCM) is atomic.



### 1.5.7.1 Clocking components

Clocking components are components with the LISA property `component_type = "clocking"`. They provide a mechanism for systems to regulate the execution rate of components.

Clocking includes the concept of clock rates, dividers to change clock rates, and timers to generate callbacks based on those clock rates.

If the [MasterClock](#) component is instantiated in a system, it provides a consistent master clock rate. Although this rate is not defined, you can consider this to be 1 Hz, even for non-SystemC systems. [ClockDivider](#) components can convert this clock rate into a new rate using a multiplier and divider, although the clock rate cannot be less than 1 Hz. You can cascade [ClockDivider](#) components to produce many different clock rates within a system. The maximum ratio of any two clocks in the system must be less than 232.

[ClockTimer](#) components can be instantiated by a component and connected to any [MasterClock](#) or [ClockDivider](#) output. [ClockTimers](#) can generate callbacks after a given number of ticks of that clock. [ClockTimers](#) can invoke a behavior on the component to permit the component to perform work. The component can then request the [ClockTimer](#) to repeat its count.

## 1.5.8 Iris interactions with processor reset behavior

Architecturally, Arm®Cortex®-M series processors have different reset behavior to that of A and R series processors.

For both hardware and model processors, a reset consists of asserting, and then deasserting the reset pin. However:

### A and R series

Asserting reset causes the processor to stop executing instructions and on deasserting reset, the processor begins fetching and executing instructions from a defined address.

### M series

The architecture documentation specifies that on asserting the reset pin, the processor stops executing instructions. On deasserting the reset pin, the VTOR is given its reset value, SP\_main is read from address VTOR+0, and the PC is read from address VTOR+4. See the Reset behavior section and the Vector Table Offset Register, VTOR section in the [ARMv7-M Architecture Reference Manual](#). Also, the processor begins fetching and executing instructions.

The reset behavior on M series processors interacts differently with Iris debugging functionality compared to A and R series processors:

### A and R series

After the reset pin of the processor is asserted, a new application can be loaded using the `image_loadFile()` call in Iris. This loads an application into memory, and sets the PC if a start address is specified in the application. When reset is deasserted, the processor begins fetching and executing from the start address as expected. Similarly, if a debugger asserts

reset on the processor, it can modify memory with a sequence of `memory_write()` calls, update the PC with a `resource_write()` call, and when reset is deasserted, the processor begins to fetch and execute from the PC.

**M series:**

These techniques do not work because after reset is deasserted, the processor updates `SP_main` and the PC, overwriting the settings made by the Iris calls.

For these techniques to be possible, the M series processor models differ slightly from the architectural reset behavior. When reset is asserted, the M series makes note of whether the PC or SP is modified before reset is next deasserted. If there are any Iris writes to the PC or SP registers, either directly through `resource_write()` or indirectly through `image_loadFile()`, this is tracked. When reset is deasserted, if the PC has been modified using Iris, the PC retains the value written to it, otherwise it reads it from address `VTOR+4`. Similarly, if the SP has been modified using Iris, `SP_main` retains the value written to it, otherwise it reads it from address `VTOR+0`.

## 1.6 syncLevel definitions

Definitions for the possible `syncLevel` values.

**syncLevel 0**

The simulator runs as fast as possible. It does not permit inspection of the processor registers while the simulation is running, and does not stop synchronously when requested to do so.

After enabling or disabling a trace source, it is undefined how many instructions are executed before the change takes effect.

Quantum end detection guarantees that a quantum is not overshoot indefinitely. Quantum end detection applies to, but is not limited to, backward branches, indirect jumps, exceptions, and atomic operation retries. In addition to temporal quantum end detection, some events may end a quantum, like executing a barrier, entering a low power state, or accessing a peripheral. Target software and simulation controllers must not rely on a specific scheduling pattern based on these quantum end check points.

Use cases: normal fast simulation and normal debugging when no watchpoint is set.

**syncLevel 1**

The simulation runs slightly slower than `syncLevel 0`. Iris can read the up-to-date values of the processor registers, including PC and instruction count. You cannot stop the simulation synchronously.

After enabling or disabling a trace source, it is undefined how many instructions will be executed before the change takes effect.

Quantum end detection is as for `syncLevel 0`.

Use cases: external breakpoints that block the simulation, inspect state of processor or memory from within a peripheral or memory access.

**syncLevel 2**

As for syncLevel 1, except that you can stop the simulation synchronously from within all `LD` or `ST` and similar instructions. The simulation stops immediately after the current `LD` or `ST` instruction has been completely executed (post instruction).

After enabling or disabling a trace source, it is likely, but not guaranteed, that the change will be visible sooner than with syncLevel 0 or 1.

Quantum end detection is as for syncLevel 1, plus it includes the end of `LD` or `ST` instructions.

Use cases: Watchpoints, external breakpoints, stopping from within `LD` or `ST`-related MTI callbacks.

**syncLevel 3**

As for syncLevel 2, except that you can stop the simulation synchronously from within any instruction. The simulation stops immediately after the current instruction has been completely executed (post instruction).

After enabling or disabling a trace source, the change is visible at the next instruction that is executed.

Quantum end detection is as for syncLevel 2. This allows switching between syncLevels 2 and 3 without changing the simulation scheduling.

Use cases: a Stop from within arbitrary MTI callbacks such as the `INST` callback. This syncLevel is a fallback for all use cases that do not fall into syncLevels 0-2.

**Related information**

[Simulation accuracy API in Iris User Guide](#)

## 1.7 Controlling and observing the syncLevel

Iris memory breakpoints automatically register and unregister for their required syncLevel (`POST_INSN_IO`). All other use cases must explicitly register and unregister for the syncLevel they require.

Users of syncLevel write to a set of non-architectural processor registers in the Iris interface to register and unregister for specific syncLevels. Registers are more suitable than parameters for exposing an interface that has side effects on writes and where values might change spontaneously.

The following list of registers is the exposed interface to control and observe the syncLevel. All these registers are in the register group `simulation` for each CT processor that contains non-architectural, simulator-specific registers. All are 32-bit integer registers and are write-only. To register and unregister for the syncLevel you require, write the value 0 to the register. Changes to the syncLevel become effective at the next stop event checkpoint. You can also write to these

registers any time before the simulation is running, for example from the `init()` simulation phase. The change takes effect immediately when the simulation is run.

**syncLevelSyncStateRegister**

Register for `SYNC_STATE`

**syncLevelSyncStateUnregister**

Unregister for `SYNC_STATE`

**syncLevelPostInsnIORegister**

Register for `POST_INSN_IO`

**syncLevelPostInsnIOUnregister**

Unregister for `POST_INSN_IO`

**syncLevelPostInsnAllRegister**

Register for `POST_INSN_ALL`

**syncLevelPostInsnAllUnregister**

Unregister for `POST_INSN_ALL`

The following registers are only for debugging and visibility in the debugger. `syncLevel` users do not usually access them. The registers marked “use as read-only” must be treated as read-only. However, you can write to them to permit debugging the `syncLevel` mechanisms. They are not memory (or CPnn-) mapped anywhere, and are not accessible to target programs:

**syncLevel**

Current `syncLevel`. This register enables you to see what kind of performance you can expect from the model. Read-only.

**syncLevelSyncStateCount**

User counter. Read/write (use as read-only).

**syncLevelPostInsnIOCount**

User counter. Read/write (use as read-only).

**syncLevelPostInsnAllCount**

User counter. Read/write (use as read-only).

**minSyncLevel**

Same as the `min_sync_level` parameter, described next. Read/write.

In addition to the debug register interface, there is a parameter that can influence the `syncLevel`:

```
min_sync_level (default=0, type=int, runtime=true)
```

This parameter enables you to control the minimum `syncLevel` using the parameter interface. This is not intended to be the primary interface to control the `syncLevel` because it does not enable multiple independent `syncLevel` users to indicate their requirements to the simulator. It is primarily for debugging purposes and for situations where a single global `syncLevel` setting is sufficient.

You can change this parameter at runtime, and changes become effective at the next stop event checkpoint. Reading this parameter value returns the `min_sync_level`, not the current `syncLevel`.

This parameter is only an additional way of controlling the `syncLevel` and controls the same mechanisms as the register interface.

## 1.8 User mode networking

User mode networking emulates a built-in IP router and DHCP server, and routes TCP and UDP traffic between the guest and host. It uses the user mode socket layer of the host to communicate with other hosts.

This allows the use of a significant number of IP network services without requiring administrative privileges, or the installation of a separate driver on the host on which the model is running. Fast Models supports the following kinds of Ethernet device models:

### SMSC\_91C111

This is paired with an external [HostBridge](#) component. The user mode networking specification is set on the external HostBridge.

### VirtioNetMMIO

This has a built-in [HostBridge](#) sub-component. The user mode networking specification is set on the internal HostBridge.



- You can use TCP and UDP over IP, but not ICMP (ping).
- User mode networking does not support forwarding UDP ports on the host to the model.
- You can only use DHCP within the private network.
- You can only make inward connections by mapping TCP ports on the host to the model. This is common to all implementations that provide host connectivity using NAT.
- Operations that require privileged source ports, for example NFS in its default configuration, do not work.
- If setup fails, or the parameter syntax is incorrect, there is no error reporting.

To enable user mode networking, run the model with the following parameters:

### SMSC\_91C111

```
-C motherboard.hostbridge.userNetworking=true  
-C motherboard.smc_91c111.enabled=true
```

### VirtioNetMMIO

```
-C motherboard.virtio_net.hostbridge.userNetworking=true  
-C motherboard.virtio_net.enabled=true
```

To map a host TCP port to a model port, run the model with the `userNetPorts` parameter. This parameter allows services to appear to be listening on privileged ports in the model but be mapped to unprivileged ports on the host. The syntax is a comma-separated list of items in the form:

[host-ip:]hostport=[model-ip:]modelport

For example, to map port 8022 on the host to port 22 on the model, use this parameter:

### SMSC\_91C111

```
-C motherboard.hostbridge.userNetPorts=8022=22
```

### VirtioNetMMIO

```
-C motherboard.virtio_net.hostbridge.userNetPorts=8022=22
```

Either or both of a host IP address and model IP address can optionally be specified on either side of the assignment to select a specific interface on which the mapping will occur. For example:

```
127.0.0.1:8022=127.0.0.1:22
```

The default is to accept connections on any interface.

## 1.9 TAP/TUN networking

This section describes Fast Models support for TAP/TUN networking.

### 1.9.1 TAP/TUN networking limitations

TAP/TUN networking on Fast Models has these limitations.

- It is only supported on Linux, not on Windows.
- If the host uses Dynamic DNS, it inserts records into DNS. If you manage this host with DHCP, installing TAP networking can cause failure to register in the DNS. After the physical device attaches to the bridge device, the DHCP client reruns, but the DHCP request does not have the correct hostname.
- Most WiFi adaptors do not implement the required support for TAP networking to work.

### 1.9.2 Setting up a network connection for Red Hat Enterprise Linux

This section describes how to set up a network connection.

#### Before you begin

Ensure that the `brctl` utility is on your system. This utility is part of the standard Linux bridge utilities, `bridge-utils`, which are in the Linux distribution.

## About this task

---



Note

- Perform this procedure once for each host machine.
  - The setup and configuration instructions assume that your network provides IP addresses by DHCP. Otherwise, consult your network administrator.
- 

### Procedure

1. In a shell, change to the `$PVLIB_HOME/ModelNetworking` directory.
2. Run the following script from this directory, because it does not work correctly if run from any other location:

#### 32-bit operating system

Run `add_adapter_32.sh` as root. For example, `sudo ./add_adapter_32.sh`.

#### 64-bit operating system

Run `add_adapter_64.sh` as root. For example, `sudo ./add_adapter_64.sh`.

3. The prompt appears: **Specify the TAP device prefix:(ARM)**. Select **Enter** to accept the default.
  4. The prompt appears: **Specify the user list**. Enter a space-separated list of all users who are to use the model on the network, then select **Enter**. All entries in the list must be the names of existing user accounts on the host.
  5. The prompt appears: **Enter the network adapter which connects to the network:(eth0)**. Select **Enter** to accept the default, or input the name of a network adapter that connects to your network.
  6. The prompt appears: **Enter a name for the network bridge to create:(armbr0)**. Select **Enter** to accept the default, or input a name for the network bridge. You must not have an existing network interface on your system with the selected name.
  7. The prompt appears: **Enter the location to write the init script to:(/etc/init.d/FMNetwork)**. Select **Enter** to accept the default, or input another path with a new filename in an existing directory.
  8. The prompt appears: **WARNING: the script creates a bridge which includes the local network adapter and tap devices. You may suffer temporary network loss. Do you want to proceed? (Yes or No)**. Verify all values input so far, and enter **Yes** if you want to proceed. If you enter **No**, no changes are made to your system.
  9. A prompt appears to inform you of the changes that the script is to make to your system. Input **Yes** if you are happy to accept these changes, or input **No** to leave your system unchanged.
- 



Note

After entering **Yes**, you might temporarily lose network connectivity. Also, the IP address of the system might change.

---

## Next steps

The network bridge is disabled after the host system is reset. To configure the host system to support bridged networking, you might have to create links to the `init` script (FMNetwork). The script suggests some appropriate links for Red Hat Enterprise Linux.

The default firewall configuration on Red Hat Enterprise Linux blocks packet transmission across the TAP networking bridge device. You can disable the firewall. If the context makes this unwise, then add firewall rules to allow transmission. These `iptables` commands configure the firewall to allow packets across the bridge device:

```
iptables -I FORWARD -m physdev --physdev-is-bridged -j ACCEPT
service iptables save
service iptables restart
```

## 1.9.3 Setting up a network connection for Ubuntu Linux

This section describes how to set up a network connection.

### About this task

To use TAP networking with Fast Models on Ubuntu, set up a TAP device manually by following these steps. This guide uses a network interface `eth0` and a username `fmuser`. Replace these values as appropriate.



Typographic errors when modifying the network configuration can cause failure to connect to the network. We recommend performing these steps on a machine that you have physical access to.

---

### Procedure

1. If it is not present, add `eth0` to the interfaces file `/etc/network/interfaces`. This step stops network-manager from managing `eth0`. It can result in network-manager indicating there is no network connection even if there is. You must have root privileges for this step.

Use one of the following ways:

- For an interface using DHCP, add:

```
auto eth0
iface eth0 inet dhcp
```

- To configure a static IP address, add the static information, for example:

```
auto eth0
iface eth0 inet static
address 192.168.0.2
netmask 255.255.255.0
gateway 192.168.0.1
```



The network notifier applet launches the GUI tool network-manager. It automatically configures network devices that `/etc/network/interfaces` does not manage, and sets up devices in a way that is incompatible with bridging. This step ensures that network-manager does not manage the network interface that you want to bridge to. If you are unsure how to configure your network interface, ask your network administrator.

2. Install the bridge-utils package:

```
sudo apt-get install bridge-utils
```

This step provides the `brctl` command for creating and managing the network bridge.

3. Create a bridge device by adding this entry to `/etc/network/interfaces`:

```
auto armbr0
iface armbr0 inet dhcp
pre-up ifconfig eth0 0.0.0.0 promisc
post-down ifconfig eth0 0.0.0.0 -promisc
```

The pre-up and post-down lines give commands to execute before bringing up `armbr0` and after bringing it down. These commands put `eth0` into promiscuous mode at pre-up and take it out of promiscuous mode at post-down. Promiscuous mode makes sure that the hardware does not filter out packets for the virtual ethernet device.

This step creates a bridge device that is called `armbr0` from Fast Models TAP devices to the physical network.

4. Create the TAP devices. TAP devices need permission for specific users, so create one for each user who is to run the model with the virtual ethernet device.  
For example, to create a TAP device called `ARMfmuser` for the user `fmuser`, add the following lines to the `armbr0` section of `/etc/network/interfaces`.

```
pre-up ip tuntap add dev ARMfmuser mode tap user fmuser
pre-up ifconfig ARMfmuser 0.0.0.0 promisc
post-down ip tuntap del dev ARMfmuser mode tap
```

This step creates a TAP device for each user.

5. Create a bridge between the TAP devices and the network interface `eth0` by adding a `bridge_ports` line to the `armbr0` section of `/etc/network/interfaces`. For example, for a TAP device that is named `ARMfmuser`, add the following line:

```
bridge_ports eth0 ARMfmuser
```

6. The added `/etc/network/interfaces` code now looks like this:

```
auto armbr0
iface armbr0 inet dhcp
pre-up ifconfig eth0 0.0.0.0 promisc
post-down ifconfig eth0 0.0.0.0 -promisc
pre-up ip tuntap add dev ARMfmuser mode tap user fmuser
pre-up ifconfig ARMfmuser 0.0.0.0 promisc
post-down ip tuntap del dev ARMfmuser mode tap
bridge_ports eth0 ARMfmuser
```

- Restart network services by either restarting the computer or by running the following commands:

```
sudo ifdown eth0 && sudo ifup eth0
sudo ifup armbr0
sudo service network-manager restart
```

**Note**

armbr0 must be explicitly started.

This step disconnects and reconnects all network interfaces.

## 1.9.4 Configuring the networking environment for Linux

This section describes how to set the parameters to make a network connection.

### About this task

**Note**

Firewall software might block network traffic in the network bridge and result in a networking failure. If the model does not work after configuration, check the firewall settings.

### Procedure

Set the parameters on the `HostBridge` and `smSc_91c111` components, or on the `virtioNetMMIO` component and its `HostBridge` subcomponent. For example:

#### SMSC\_91C111:

```
hostbridge.interfaceName=ARM<username>
smSc_91c111.enabled=1
```

#### VirtioNetMMIO:

```
virtio_net.hostbridge.interfaceName=ARM<username>
virtio_net.enabled=1
```

ARM<username> is an adapter that is built into the network bridge.

### Related information

[Fast Models Tools User Guide](#)

## 1.9.5 Solutions to networking issues on Linux

This section describes how to solve networking issues.

### The model networking works after initial setup, but stops working after reboot

Set the correct access permissions for the `/dev/net/tun` device, by executing `chmod 666 /dev/net/tun` as root. To preserve the change across reboots, modify the udev rules of the TAP device by opening `/etc/udev/rules.d/50-udev.rules` as root, and finding the line:

```
KERNEL=="tun", NAME="net/%k"
```

If it does not have `MODE="0666"` at the end of the line, append `MODE="0666"`:

```
KERNEL=="tun", NAME="net/%k", MODE="0666"
```

### Model networking installs correctly, but when a model starts up, the model cannot receive packets

Disable the firewall on the host machine, or add the TAP device to `trusted devices`.



Note

Refer to the vendor's documentation manual.

---

## 1.9.6 Disabling and re-enabling networking for Linux

This section describes how to disable and re-enable networking with an `init` script.

### About this task



Caution

These operations remove/restore TAP devices and the network bridge. There is a temporary loss of network connectivity and your IP address might change.

---

### Procedure

1. To disable networking without uninstalling it, invoke the installed `init` script (by default, `/etc/init.d/FMNetwork`) as root with the parameter `stop`:

```
sudo /etc/init.d/FMNetwork stop
```

2. To re-enable networking, invoke the `init` script as root with the parameter `start`:

```
sudo /etc/init.d/FMNetwork start
```

## 1.9.7 Uninstalling networking for Linux

This section describes the steps to uninstall a network.

### Procedure

1. In a shell, change to the `$PVLIB_HOME/ModelNetworking/` directory.
2. Run `uninstall.sh` as root, passing the location of the `init` script (FMNetwork):

```
sudo ./uninstall.sh /etc/init.d/FMNetwork
```

You must run this script from the directory in which it is installed, because it does not work correctly if run from any other location.



There is a temporary loss of network connectivity and your IP address might change.

### Next steps

The uninstall script removes everything that can be safely removed. It does not remove:

- symlinks to the `init` script. You must remove any symlinks that you have created.
- `/sbin/brcctl`. Removing this is optional.

## 1.10 Using parameters to set port values

Some processor and peripheral component ports are almost always static in value when used as part of a typical platform. For example, the reset vector base address register address (RVBARADDR) port in processor components.

To facilitate easy configuration of platform models, the IP models for these components can provide a shadow parameter for these ports. This parameter can be used to change the value that is used by the model. In these cases, the following rules apply:

- If a port is driven in the platform model, then the parameter value is ignored.
- If a port is not driven in the platform model, then the parameter value is sampled at both simulator reset, and at every subsequent simulation reset of the specific IP model.



Simulator reset corresponds with the LISA `reset()` behavior and the SystemC `start_of_simulation()` callback.

- All ports and parameters that are sampled at reset are sampled when the simulation reset signal concerned is deasserted.

- If a port is not driven in the platform model, and a parameter has not been set, then the default value for the parameter is used.

In some IP models, the value of some ports can only be set by using a parameter. That is, the parameter is provided instead of the port.

## 1.11 PVBUS C++ transaction and Tx\_Result classes

This section describes the C++ transaction and `Tx_Result` classes.

### 1.11.1 Class `pv::TransactionGenerator`

This class provides efficient mechanisms for bus masters to generate transactions that are transmitted over the `pvbustm` port of the associated PVBUSMaster subcomponent.

You can produce `pv::TransactionGenerator` objects by invoking the `createTransactionGenerator()` method on the control port of a PVBUSMaster component.

```
class pv::TransactionGenerator
{
    // Tidy up when TransactionGenerator is deleted.
    ~TransactionGenerator()

    // Control AXI-specific signal generation for future transactions.
    // Privileged processing mode.
    void setPrivileged(bool priv = true);

    // Instruction access (vs data).
    void setInstruction(bool instr = true);

    // Normal-world access (vs secure).
    void setNonSecure(bool ns = true);

    // Locked atomic access.
    void setLocked(bool locked = true);

    // Exclusive atomic access.
    void setExclusive(bool excl = true);

    // Generate transactions.
    // Generate a read transaction.
    bool read(bus_addr_t, pv::AccessWidth width, uint32_t *data);

    // Generate a write transaction.
    bool write(bus_addr_t, pv::AccessWidth width, uint32_t const *data);

    // Generate read transactions.
    bool read8(bus_addr_t, uint8_t *data);
    bool read16(bus_addr_t, uint16_t *data);
    bool read32(bus_addr_t, uint32_t *data);
    bool read64(bus_addr_t, uint64_t *data);

    // Generate write transactions.
    bool write8(bus_addr_t, uint8_t const *data);
    bool write16(bus_addr_t, uint16_t const *data);
    bool write32(bus_addr_t, uint32_t const *data);
    bool write64(bus_addr_t, uint64_t const *data);
};
```

### 1.11.2 TransactionGenerator efficiency considerations

TransactionGenerators are most efficient for multiple accesses to one 4KB page.

Each TransactionGenerator caches connection information internally. This improves efficiency for multiple accesses to a single 4KB page. If a component requires repeated access data from different pages, for example when streaming from one location to another, we recommend you create a TransactionGenerator for each location.

You can dynamically create and destroy TransactionGenerators, but it is better to allocate them once at initialization and destroy them at shutdown. See the example in `$PVLIB_HOME/examples/LISA/BusComponents/DmaTransfer.lisa`.

### 1.11.3 Enum `pv::AccessWidth`

This enum selects the required bus width for a transaction.

Defined values are:

- `pv::ACCESS_8_BITS`
- `pv::ACCESS_16_BITS`
- `pv::ACCESS_32_BITS`
- `pv::ACCESS_64_BITS`

### 1.11.4 Class `pv::Transaction`

This is a base class for read and write transactions that are visible in the `PVBusSlave` subcomponent. It contains functionality common to both types of transaction.

It provides an interface that permits bus slaves to access the details of the transaction. Do not instantiate these classes manually. The classes are generated internally by the `PVBus` infrastructure.

This base class provides access methods to get the transaction address, access width, and bus signals. It also provides a method to signal that the transaction has been aborted.

```
class pv::Transaction
{
public:
    // Accessors
    bus_addr_t      getAddress() const;           // Transaction address
    bus_addr_t      getAddressEndIncl() const;    // Address of last byte in the
transaction
    bus_addr_range_t getAddressRange() const;    // The range of addresses that the
transaction accesses
    void            setAddress(bus_addr_t);       // Transaction address.

    AccessWidth     getAccessWidth() const;       // Request width in lg2 bytes
    int             getAccessByteWidth() const;   // Request width in bytes
    int             getAccessBitWidth() const;    // Request width in bits
}
```

```

    bool        isAligned() const;           // Request address is aligned to
    request width boundary.

    bool        isPrivileged() const;        // Privileged process mode?
    bool        isInstruction() const;        // Instruction request vs data?
    bool        isNonSecure() const;         // Normal-world vs secure-world?
    PASpace_t   getPhysicalAddressSpace() const; // PhysicalAddressSpace of access
    bool        isLocked() const;            // Atomic locked access?
    bool        isExclusive() const;         // Atomic exclusive access?
    bool        isCacheMaintenance() const;  // Cache maintenance operation
    bool        isWithoutData() const;       // Transaction has no data payload
    uint64_t    getManagerID64() const;      // Transaction manager ID 64
    bool        hasSideEffect() const;       // The transaction has side effect?

    // Generate transaction returns
    Tx_Result   generateAbort();              // Cause the transaction to abort
    Tx_Result   generateSlaveAbort();         // Cause the transaction to abort
    Tx_Result   generateDecodeAbort();       // Cause the transaction to abort
    Tx_Result   generateExclusiveAbort();    // Cause the transaction to abort

    Tx_Result   resetOccurred();             // Indicate that a reset occurred
};

```

### 1.11.5 Class pv::ReadTransaction

This class extends the pv::Transaction class to provide methods for returning data from a bus read request.

```

class ReadTransaction : public Transaction
{
public:
    /*! Return a 64-bit value on the bus. */
    Tx_Result setReturnData64(uint64_t);

    /*! Return a 32-bit value on the bus. */
    Tx_Result setReturnData32(uint32_t);

    /*! Return a 16-bit value on the bus. */
    Tx_Result setReturnData16(uint16_t);

    /*! Return an 8-bit value on the bus. */
    Tx_Result setReturnData8(uint8_t);

    /*! This method provides an alternative way of returning a Tx_Result
     * success value (instead of just using the value returned from
     * setReturnData<n>()).
     *
     * This method can only be called if one of the setReturnData<n>
     * methods has already been called for the current transaction.
     */
    Tx_Result readComplete();
};

```

### 1.11.6 Class pv::WriteTransaction

This class extends the pv::Transaction class to provide methods for returning data from a bus write request.

```

class WriteTransaction : public Transaction
{

```

```

public:
    /*! Get bottom 64-bits of data from the bus. If the transaction width
    * is less than 64-bits, the data is extended as appropriate.
    */
    uint64_t getData64() const;

    /*! Get bottom 32-bits of data from the bus. If the transaction width
    * is less than 32-bits, the data is extended as appropriate.
    */
    uint32_t getData32() const;

    /*! Get bottom 16-bits of data from the bus. If the transaction width
    * is less than 16-bits, the data is extended as appropriate.
    */
    uint16_t getData16() const;

    /*! Get bottom 8-bits of data from the bus. If the transaction width
    * is less than 8-bits, the data is extended as appropriate.
    */
    uint8_t getData8() const;

    /*! Signal that the slave has handled the write successfully.
    */
    Tx_Result writeComplete();
};

```

## 1.12 Visualisation library

The Visualisation library does not model hardware directly but instead provides components, protocols, and a library. These permit a GUI display that lets you interact with the external I/O from the model platform.

The types of I/O handled include:

- LCD display, such as the output from the PL110\_CLCD component display port.
- LEDs representing values from a ValueState port as either single lights, or as segmented alphanumeric displays.
- DIP switches, which can drive a ValueState port.
- Capture of keyboard and mouse input, using the KeyboardStatus and MouseStatus protocols to feed input to a PS2Keyboard or PS2Mouse component.
- Background graphics, custom rendered graphics, and clickable push buttons, permitting the UI to display a skin representing the physical appearance of the device being modeled.
- Status information such as processor instruction counters, with values taken from the InstructionCount port of a processor.

The Visualisation library provides a C++ API that enables you to write your own visualization components in LISA+. These custom components can display any combination of the supported I/O types.

You can add the prebuilt GUIPoll component to your custom component. The GUIPoll component provides a LISA visualization component with a periodic signal that keeps the display updated, even when the simulation is stopped.



The Visualisation library supports one signaling protocol, the LCD protocol.

### Related information

[LCD protocol](#) on page 100

[LISA visualisation models](#) on page 53

[Visualisation library C++ classes](#) on page 53

## 1.12.1 LISA visualisation models

The visualisation components provide a host window to display status information in addition to a frame buffer.

Each example platform model contains its own LISA visualisation model. You can use the model as the basis for your own visualization-containing components, such as the PL110\_CLCD component. To use the visualisation components in your own system, copy the component LISA files from the relevant platform model directory, because they are not in the generic model library.

## 1.12.2 Visualisation library C++ classes

This section describes the C++ classes and structures in the Visualisation library.

### 1.12.2.1 C++ classes inclusion

To use these Visualisation library classes, begin your LISA component with the correct `#include` statement.

```
includes
{
#include "components/Visualisation.h"
}
```

### 1.12.2.2 Class Visualisation

The `visualisation` class is the API for creating a custom LISA visualization component.

A component obtains an instance of this class by calling the global function `createVisualisation()`. The component can then use this instance to control the size and layout of the visualization window:

#### **Visualisation \*createVisualisation()**

This function generates an instance of the Visualisation library. You can only call this function once, because SDL only supports opening a single display window. The Visualisation library is implemented using the Simple DirectMedia Layer (SDL) cross-platform rendering library.

The `visualisation` class has the following methods:

**~Visualisation()**

Destructor for the Visualisation library. You must only call this method when your simulation is shutting down, after all allocated resources (VisRenderRegions, VisPushButtonRegions, VisBitmaps) have been deleted.

**void configureWindow(unsigned int width, unsigned int height, unsigned int bit\_depth)**

Sets the visualization window to the requested size and bit depth. Depending on the display capabilities, the window might actually get a different bit depth from the size you requested.

**VisBitmap \*loadImage(char const \*filename)**

Allocates a new VisBitmap object, initialized by loading a Microsoft Windows Bitmap (.BMP) from the given file.

**VisBitmap \*loadImageWithAlphaKey(char const \*filename, unsigned int red, unsigned int green, unsigned int blue)**

Allocates a VisBitmap object, as with `loadImage()`. All pixels of the color specified by `red`, `green`, `blue` are converted into a transparent alpha channel.

**VisBitmap \*cropImage(VisBitmap \*source, int x, int y, unsigned int width, unsigned int height)**

Allocates a new VisBitmap object, by cropping a region from the source bitmap.

**void releaseImage(VisBitmap \*)**

Releases the resources held by the given VisBitmap. The underlying bitmap is only to be unloaded if it is not in use.

**void setBackground(VisBitmap \*background, int x, int y)**

Sets the background image for the visualization window. This takes a copy of the data referenced by the VisBitmap, so it is safe for the client to call `releaseImage(background)` immediately after calling `setBackground()`. The background is not displayed until the first call to `poll()`.

**VisRenderRegion \*createRenderRegion()**

Allocates a new VisRenderRegion object that can be used to display arbitrary graphics, including LCD contents, in a rectangular region.

**VisPushButtonRegion \*createPushButtonRegion()**

Allocates a new VisPushButtonRegion, which can be placed at a location on the display to provide a clickable push button.

**bool poll(VisEvent \*event)**

Permits the Visualisation library to poll for GUI events. The client passes a reference to a VisEvent structure, which receives details of a single mouse/keyboard event.

The method returns false if no events have occurred.

Your LISA visualization implementations must call this periodically by using a GUIPoll component. On each `gui_callback()` event, you must ensure that the visualization component repeatedly calls `poll()` until it returns false.

**void lockMouse(VisRegion \*region)**

Locks the mouse to the visualization window and hides the mouse pointer.

**void unlockMouse()**

Unlocks and redisplay the mouse pointer.

**bool hasQuit()**

Returns true if the user has clicked on the close icon of the visualization window.

### 1.12.2.3 Class VisRegion

This class is the common base class for VisPushButtonRegion and VisRenderRegion, representing a region of the visualization display.

**~VisRegion()**

Permits clients to delete a VisPushButtonRegion when it is no longer required.

**void setId(void \*id)**

Permits a client-defined identifier to be associated with the region.

**void \*getId()**

Returns the client-defined identifier.

**void setVisible(bool vis)**

Specifies whether the region is to be displayed on the screen. This is currently ignored by the SDL implementation.

**void setLocation(intx, int y, unsigned int width, unsigned int height)**

Sets the location of this region relative to the visualization window.

### 1.12.2.4 Class VisPushButtonRegion : public VisRegion

This class defines a region of the visualization window that represents a clickable button.

Optionally, the button can provide different VisBitmap representations for a button-up and a button-down graphic, and a graphic to use when the mouse pointer rolls over the button.

In addition to the public method defined in VisRegion, this class defines these methods:

- `void setButtonUpImage(VisBitmap*bmpUp) : void`
- `setButtonDownImage(VisBitmap*bmpDown) : void`
- `setButtonRollOverImage(VisBitmap*bmpRollover)`

These methods set the graphics to be used for each of the button states. If any image is not specified or is set to NULL, then the corresponding area of the visualization background image is used.

The VisPushButtonRegion takes a copy of the VisBitmap, so the client can safely call `Visualisation::releaseBitmap()` on its copy.

- `void setKeyCode(intcode)`

This method sets the code for the keypress event that is generated when the button is pressed or released.

### 1.12.2.5 Class `VisRenderRegion` : `public VisRegion`

This class defines a region of the visualization window that can render client-drawn graphics, including a representation of the contents of an LCD.

In addition to the public method defined in `VisRegion`, the class defines these methods:

**`VisRasterLayout const *lock()`**

Locks the region for client rendering. While the buffer is locked, the client can modify the pixel data for the buffer. You must not call the methods `writeText()` and `renderBitmap()` while the buffer is locked.

**`void unlock()`**

Releases the lock on the render buffer, permitting the buffer to be updated on screen.

**`void update(int left, int top, unsigned int width, unsigned int height)`**

Causes the specified rectangle to be drawn to the GUI.

**`int writeText(const char *text, int x, int y)`**

Renders the given ASCII text onto an unlocked `VisRenderRegion`. The return value is the x co-ordinate of the end of the string. The default font is 8 pixels high, and cannot be changed.

**`void renderBitmap(VisBitmap *bitmap, int x, int y)`**

Draws a bitmap onto an unlocked `VisRenderRegion`.

### 1.12.2.6 Struct `VisRasterLayout`

This struct defines the layout of the pixel data in a frame-buffer.

The `lock()` method of the LCD protocol expects to be given a pointer to this structure. You can generate a suitable instance by calling `VisRasterRegion::lock()`.

The structure contains these fields:

**`uint8_t* buffer`**

This points to the buffer for the rasterized pixel data. The controller can write pixels into this buffer, but must stay within the bounds specified by the width and height.

**`uint32_t pitch`**

The number of bytes between consecutive raster lines in the pixel data. This can be greater than the number of bytes per line.

**`uint32_t width`**

The width, in pixels, of the render area. This value can be less than the width requested by the LCD controller when it called `lock()`.

**uint32\_t height**

The height, in pixels, of the render area. This value can be less than the height requested by the LCD controller when it called `lock()`.

**VisPixelFormat format**

This structure defines the format of the pixel data in the raster buffer.

**bool changed**

This is set to true if the pixel format or buffer size has changed since the previous call to `lock()`.

Pixel data is represented as a one-dimensional array of bytes. The top-left pixel is pointed to by the buffer member. Each pixel takes up a number of bytes, given by `format.pbytes`.

The pixel at location (x, y) is stored in the memory bytes starting at:

```
buffer[y * pitch + x * format.pbytes]
```

### 1.12.2.7 Struct VisPixelFormat

This struct specifies the format of pixel data within the buffer.

The members are:

**uint32\_t rbits, gbits, bbits**

The number of bits per color channel.

**uint32\_t roff, goff, boff**

The offset within the pixel data value for the red/green/blue channels.

**uint32\_t pbytes**

The size of a single pixel, in bytes.

`format.pbytes` specifies the number of bytes that make up the data for a single pixel. These bytes represent a single pixel value, stored in host-endian order. The pixel value contains a number of the form:

```
(R<<format.roff) + (G<<format.goff) + (B<<format.boff)
```

where (R,G,B) represents the values of the color channels for the pixel, containing values from 0 up to  $(1<<\text{format.rbits})$ ,  $(1<<\text{format.gbits})$ ,  $(1<<\text{format.bbits})$ .

## 1.13 Timing annotation

Timing annotation enables you to perform high-level performance estimation on Fast Models.

Fast Models are Programmers View (PV) models that sacrifice timing accuracy to achieve fast simulation execution speeds. By default, each instruction takes a single simulator clock cycle, with no delays for memory accesses.

Timing annotation enables you to perform more accurate performance estimation on Fast Models with minimal simulation performance impact. You can use it to show performance trends and to identify test cases for further analysis on approximately timed or cycle-accurate models.

Timing annotation is always enabled for Fast Models platforms.

You can configure the following aspects of timing annotation:

- The time that processors take to execute instructions. This can be modeled in either of the following ways:
  - As an average Cycles Per Instruction (CPI) value, using the `cpi_mul` and `cpi_div` model parameters. [Timing annotation](#) in Fast Models Tutorials shows how to use this feature and measure its impact on code execution time.
  - By assigning CPI values to different instruction classes, using [CPI files](#).
- [Instruction and data prefetching](#).
- [Cache and TLB latency](#).

### 1.13.1 CPI files

Cycles Per Instruction (CPI) files define instruction classes and mapping attributes that affect timing annotation in Fast Models. They enable finer control of simulated cycle counts.

Arm does not provide CPI files, only some pre-defined CPI instruction classes which can help you to create your own CPI files. To create a CPI file for a specific CPU:

1. Create a set of mappings between the instruction encodings for the instruction set and a set of instruction classes or groups of classes. Arm provides pre-defined instruction classes and groups in `$PVLIB_HOME/etc/CPIPredefines/` for the A32, T32, and A64 instruction sets. You can include these pre-defined instruction classes in your CPI files, or you can define your own classes.
2. Create a file to map these instruction classes to CPI values. This is the CPI file. Calculate the CPI values to use based on observations from a cycle-accurate model, or see the Arm Software Optimization Guides on [Arm Developer](#).



Note

- An alternative to using CPI files is to use the `cpi_mul` and `cpi_div` core parameters. These are integers that represent a CPI multiplication or division factor for all instructions. They can be used together to represent non-integer values. For example, use `cpi_mul = 5`, `cpi_div = 4` for a CPI of 1.25.

- To calculate values for `cpi_mul` and `cpi_div`, experiment with running a workload on a cycle-accurate simulation to choose values that give the most accurate results.
- If a CPI file is present, it overrides the `cpi_mul` and `cpi_div` parameters.
- If you do not set these parameters and do not specify a CPI file, a CPI value of 1.0 is used for all instructions.

---

To specify a CPI file, either:

- Use the `--cpi-file` command-line parameter when launching the model, for example:

```
./isim_system ... --cpi-file CPI_file --stat
```

The `--stat` parameter can be used to display timing statistics on simulation exit.

- Call the `scx::scx_set_cpi_file()` function in your SystemC code.

To validate a CPI file, use the `CPIValidator` command-line tool in `$MAXCORE_HOME/bin/`. To list the available options, use the `--help` switch. For example, the following command parses and builds the evaluation tree for `CPI_file` and prints it in plain text to a file called `CPIEvaluationTree.txt`:

```
./CPIValidator --input-file CPI_file --output-file CPIEvaluationTree.txt
```

For more information, see:

- [CPI file syntax](#)
- [BNF specification for CPI files](#)

### 1.13.1.1 CPI file syntax

CPI files are plain text files that contain a series of statements, one per line. Lines that begin with a `#` character are ignored.

In the following syntax definitions, square brackets `[]` indicate optional attributes and ellipses `...` indicate attributes that can be repeated.

The valid statements in a CPI file are:

#### **DefineCpi**

Defines the CPI value to use for an instruction class or group. The syntax is:

```
DefineCpi class_or_group ISet=iset [CpuType=cputype] Cpi=cpi
where:
```

#### **class\_or\_group**

The name of an instruction class or group. This name can contain wildcards.

A decoded instruction is matched against all `DefineCpi` statements in the order they appear in the CPI file from top to bottom. The first instruction class match is used and all following statements are ignored.

**ISet=iset**

The instruction set this CPI value refers to. One of `A32`, `A64`, `Thumb`, or `T2EE`, or use the `*` character to specify all instruction sets.

**CpuType=cputype**

Specifies which Arm processor type this CPI value refers to. This parameter can be a user-defined type, or one of the following pre-defined types:

- `ARM_Cortex-A12`
- `ARM_Cortex-A17`
- `ARM_Cortex-A15`
- `ARM_Cortex-A7`
- `ARM_Cortex-A5MP`
- `ARM_Cortex-M4`
- `ARM_Cortex-M7`
- `ARM_Cortex-A57`
- `ARM_Cortex-A72`
- `ARM_Cortex-A53`
- `ARM_Cortex-R7`
- `ARM_Cortex-R5`
- `ARM_Cortex-A9MP`
- `ARM_Cortex-A9UP`
- `ARM_Cortex-A8`
- `ARM_Cortex-R4`
- `ARM_Cortex-M3`
- `ARM_Cortex-M0+`
- `ARM_Cortex-M0`

Use the `*` character to specify any processor type. Specifying no `cpuType` is equivalent to specifying `cpuType=*`.

**Cpi=cpi**

The CPI value to assign to this instruction class or group.

For example:

```
DefineCpi Load_instructions ISet=A64 CpuType=ARM_Cortex-A53 Cpi=2.15
```



**DefineClass**

Defines an instruction class. The syntax is:

```
DefineClass class Mask=mask Value=value [ProhibitedMask=pmask
ProhibitedValue=pvalue ...] ISet=iset [CpuType=cputype]
where:
```

**class**

The name of the instruction class to define. It must be unique in the CPI file. It can be used in a subsequent `DefineCpi` statement.

**Mask=mask**

A bitmask to apply to an instruction encoding before comparing the result with the `value` attribute. This parameter identifies which bits in the encoding are relevant for comparing with `value`.

For example, the value `0000xxxx1xxx100x` is represented as `Mask=0xF08E`  
`Value=0x0088`.

**Value=value**

The binary value to compare with the instruction encodings. A match indicates that the instruction belongs to this class, unless the encoding also matches the `ProhibitedValue`.

**ProhibitedMask=pmask**

A bitmask to apply to an instruction encoding before comparing the result with the `ProhibitedValue` attribute. It identifies which bits in the encoding are relevant for comparing with `ProhibitedValue`.

**ProhibitedValue=pvalue**

The binary value to compare with the instruction encodings. A match indicates that the instruction does not belong to this class.

**ISet=iset**

Specifies which instruction set this class refers to. See `DefineCpi` for the possible values.

**CpuType=cputype**

Specifies which Arm processor type this class refers to. See `DefineCpi` for the possible values.

:::note A `DefineClass` statement must include a single `Mask` and `value` attribute pair, but can include any number of `ProhibitedMask` and `ProhibitedValue` attribute pairs. :::

For example:

```
DefineClass Media_instructions Mask=0x0E000010 Value=0x06000010
ProhibitedMask=0xF0000000 ProhibitedValue=0xF0000000 ISet=A32
```

**DefineGroup**

Defines a group of instruction classes. The syntax is:

```
DefineGroup group Classes=class[,class,...] ISet=iset [CpuType=cputype]
[Mix=mix[,mix,...]]
```

where:

**group**

The name of the group to define. It must be unique in the CPI file. It can be used in a subsequent `DefineCpi` statement.

**Classes=class[,class,...]**

A comma-separated list of instruction classes that belong to this group.

**ISet=iset**

Specifies which instruction set this group refers to. See `DefineCpi` for the possible values.

**CpuType=cputype**

Specifies which Arm processor type this group refers to. See `DefineCpi` for the possible values.

**Mix=mix[,mix,...]**

A comma-separated list of mixin names that cause additional instruction groups and classes to be automatically defined.

For example:

```
DefineGroup Divide_instructions Classes=SDIV,UDIV CpuType=ARM_Cortex-A73 ISet=A32
```

**DefineMixIn**

Defines a single mask/value pair and suffix that can optionally be used in `DefineGroup` statements to automatically define new instruction groups and classes. Applying a mixin to a group causes a new instruction group or class to be defined for every instruction group or class that is included in the group, and also for the group itself. The names of these newly-defined groups and classes is the original group or class name followed by an underscore character, then the mixin suffix.

The syntax is:

```
DefineMixIn mix Mask=mask Value=value Suffix=suffix
```

where:

**mix**

The name of the mixin to define. It must be unique in the CPI file. It can be used in subsequent `DefineGroup` statements.

**Mask=mask**

A bitmask to apply to an instruction encoding before comparing the result with the `value` attribute.

**Value=value**

The binary value to compare with the instruction encodings. A match indicates that the instruction belongs to this group or class.

**Suffix=suffix**

After applying a mixin to a group, this suffix is appended to the names of the automatically-defined groups and classes.

In the following example, the `DefineGroup` statement defines `my_group`, but also automatically defines `my_group_AL` and `my_class_AL`:

```
DefineMixin my_mixin Mask=0xF0000000 Value=0xE0000000 Suffix=AL
...
DefineClass my_class Mask=0xFF000000 Value=0x03000000 ISet=A32
DefineGroup my_group Classes=my_class ISet=A32 Mix=my_mixin
```

**DefineCpuType**

Defines a processor type. The syntax is:

```
DefineCpuType cputype ISets=iset[,iset,...]
where:
```

**cputype**

The name of the processor type to define. It must be unique in the CPI file. It can be used in subsequent `DefineCpi`, `DefineClass`, `DefineGroup`, and `MapCpu` statements.

**ISets=iset[,iset,...]**

A comma-separated list of instruction sets that this processor type supports. See `DefineCpi` for the possible values.

For example:

```
DefineCpuType ARM_Cortex-A73 ISets=*
```

**MapCpu**

Maps a CPU instance by name to a CPU type. The syntax is:

```
MapCpu cpuinstance ToCpuType=cputype
where:
```

**cpuinstance**

The name of the CPU instance to map to a processor type. It can contain wildcards.

**ToCpuType=cputype**

The processor type to map the CPU instance onto. See the list of `cputypes` in `DefineCpi` for the possible values.

For example:

```
MapCpu FVP_Base_AEMvA_AEMvA.cluster0.cpu0 ToCpuType ARM_Cortex-A73
```

**Defaults**

Defines the default CPI value to be used for instructions that do not match any class or group. This statement is optional and can occur more than once in the CPI file. The syntax is:

```
Defaults ISet=iset [CpuType=cputype] Cpi=cpi
```

where:

**ISet=iset**

Specifies which instruction set this value refers to. See `DefineCpi` for the possible values.

**CpuType=cputype**

Specifies which Arm processor type this value refers to. See `DefineCpi` for the possible values.

**Cpi=cpi**

The default CPI value for the specified instruction set and processor type.

For example:

```
Defaults ISet=* CpuType=* Cpi=0.82
```

**Include**

Includes a supplementary CPI file at this point in the file. This is equivalent to the `#include` preprocessor directive in C. The evaluation of the `FilePath` attribute is to first treat it as an absolute path, then as a relative path, and finally as relative to the `PVLIB_HOME` environment variable. The syntax is:

```
Include FilePath=path
```

For example:

```
Include FilePath=etc/CPIPredefines/ARMv8A_A32_Mnemonics.txt
```

For more information, see:

- [CPI files](#)
- [BNF specification for CPI files](#)

### 1.13.1.2 BNF specification for CPI files

CPI files have the following BNF specification.

```

    <CPIFile> ::= <Statements>
    <Statements> ::= <Statement> <Statements>
                  | <Statement>
    <Statement> ::= <Comment>
                  | <DefineCpiStatement>
                  | <DefaultsStatement>
                  | <DefineCpuTypeStatement>
                  | <MapCpuStatement>
                  | <DefineClassStatement>
                  | <DefineGroupStatement>
                  | <IncludeStatement>
                  | <DefineMixInStatement>
    <DefineCpiStatement> ::= "DefineCpi" <InstructionClassOrGroup>
    <DefineCpiAttributes> <EOL>
    <DefaultsStatement> ::= "Defaults" <DefineCpiAttributes> <EOL>
    <DefineCpuTypeStatement> ::= "DefineCpuType" <UserCpuType>
    <DefineCpuTypeAttributes> <EOL>
    <MapCpuStatement> ::= "MapCpu" <CpuInstance> <MapCpuAttributes> <EOL>

```

```

<DefineClassStatement ::= "DefineClass" <InstructionClass>
<DefineClassAttributes> <EOL>
  <DefineGroupStatement ::= "DefineGroup" <InstructionGroup>
<DefineGroupAttributes> <EOL>
  <IncludeStatement> ::= "Include" <IncludeAttributes> <EOL>
  <DefineMixInStatement> ::= "DefineMixIn" <MixInType> <DefineMixInAttributes>
<EOL>
  <DefineCpiAttributes> ::= <DefineCpiAttribute> <DefineCpiAttributes>
    | <DefineCpiAttribute>
  <DefineCpiAttribute> ::= <ISetAttribute> { Mandatory }
    | <CpuTypeAttribute> { Optional }
    | <CpiAttribute> { Mandatory }
  <ISetAttribute> ::= "ISet" "=" <ISetOrStar>
  <ISetOrStar> ::= <ISet> | "*"
  <ISet> ::= "A32" | "A64" | "Thumb" | "T2EE"
  <CpuTypeAttribute> ::= "CpuType" "=" <CpuType>
  <CpuType> ::= "ARM_Cortex-A12" | "ARM_Cortex-A17"
    | "ARM_Cortex-A15" | "ARM_Cortex-A7"
    | "ARM_Cortex-A5MP" | "ARM_Cortex-M4"
    | "ARM_Cortex-M7" | "ARM_Cortex-A57"
    | "ARM_Cortex-A72" | "ARM_Cortex-A53"
    | "ARM_Cortex-R7" | "ARM_CortexR5"
    | "ARM_Cortex-A9MP" | "ARM_Cortex-A9UP"
    | "ARM_Cortex-A8" | "ARM_Cortex-R4"
    | "ARM_Cortex-M3" | "ARM_Cortex-M0+"
    | "ARM_Cortex-M0" | <UserCpuType> | "*"
  <CpiAttribute> ::= "Cpi" "=" <Cpi>
  <DefineCpuTypeAttributes> ::= <ISetsAttribute>
  <ISetsAttribute> ::= "ISets" "=" <ISetsOrStar>
  <ISetsOrStar> ::= <ISets> | "*"
  <ISets> ::= <ISet> "," <ISets> | <ISet>
  <MapCpuAttributes> ::= <ToCpuTypeAttribute>
  <ToCpuTypeAttribute> ::= "ToCpuType" "=" <CpuType>
  <DefineClassAttributes> ::= <DefineClassAttribute> <DefineClassAttributes>
    | <DefineClassAttribute>
  <DefineClassAttribute> ::= <MaskAttribute> { Mandatory }
    | <ValueAttribute> { Mandatory }
    | <ProhibitedPairsAttribute> { Optional }
    | <ISetAttribute> { Mandatory }
    | <CpuTypeAttribute> { Optional }
  <MaskAttribute> ::= "Mask" "=" <Mask>
  <ValueAttribute> ::= "Value" "=" <Value>
  <ProhibitedPairsAttribute> ::= <ProhibitedPairAttribute> <ProhibitedPairsAttribute>
    | <ProhibitedPairAttribute>
  <ProhibitedPairAttribute> ::= <ProhibitedMaskAttribute> <ProhibitedValueAttribute>
  <ProhibitedMaskAttribute> ::= "ProhibitedMask" "=" <Mask>
  <ProhibitedValueAttribute> ::= "ProhibitedValue" "=" <Value>
  <DefineGroupAttributes> ::= <DefineGroupAttribute> <DefineGroupAttributes>
    | <DefineGroupAttribute>
  <DefineGroupAttribute> ::= <ClassesAttribute> { Mandatory }
    | <ISetAttribute> { Mandatory }
    | <CpuTypeAttribute> { Optional }
    | <MixAttribute> { Optional }
  <ClassesAttribute> ::= "Classes" "=" <InstructionClassOrGroups>
  <MixAttribute> ::= "Mix" "=" <MixInTypes>
  <InstructionClassOrGroups> ::= <InstructionClassOrGroup> ",",
  <InstructionClassOrGroups>
    <InstructionClasses> ::= <InstructionClass>
  <InstructionClassOrGroup> ::= <InstructionClass>
    | <InstructionGroup>
  <MixInTypes> ::= <MixInType> ",", <MixInTypes>
  <MixInType> ::= <Symbol>
  <IncludeAttributes> ::= <FilePathAttribute>
  <FilePathAttribute> ::= "FilePath" "=" <FilePath>
  <DefineMixInAttributes> ::= <DefineMixInAttribute> <DefineClassAttributes>
  <DefineMixInAttribute> ::= <MaskAttribute>
    | <ValueAttribute>
    | <SuffixAttribute>
  <SuffixAttribute> ::= "Suffix" "=" <String>
  <FilePath> ::= <String>
  <InstructionClass> ::= <Symbol>

```

```

<InstructionGroup> ::= <Symbol>
<UserCpuType> ::= <Symbol>
<CpuInstance> ::= <QuotedString>      { Supports use of wild cards }
<Cpi> ::= <Double>
<Mask> ::= <UnsignedInteger>
<Value> ::= <UnsignedInteger>

```

For more information, see:

- [CPI files](#)
- [CPI file syntax](#)

### 1.13.2 Instruction and data prefetching

Arm® Cortex®-A series processors implement prefetching instructions and data into caches to improve cache hit rate and performance. Fast Models supports prefetching instructions and data independently, via model parameters.

For more information, see:

- [Configuring instruction prefetching](#)
- [Configuring data prefetching](#)

#### 1.13.2.1 Configuring instruction prefetching

Configure instruction cache prefetching by using the following cluster-level parameters.

**Table 1-2: Instruction prefetching parameters**

Parameter	Values	Description
icache-prefetch_enabled	true or false	Enable simulation of instruction cache prefetching; default <code>false</code> . When <code>true</code> , the other parameters in this table are available.
icache-prefetch_level	integer (zero-indexed)	Cache level to prefetch instructions into; default 0 (L1).
icache-nprefetch	integer	Number of additional sequential instruction cache lines to prefetch; default 1.



Note

These parameters only have an effect when cache state modeling is enabled.

### Usage

This example command line enables instruction cache prefetching and prints `WAYPOINT` trace events:

```

./FVP_Base_AEMvA \
-C cache_state_modelled=1 \
-C cluster0.icache-prefetch_enabled=1 \
--plugin "$PVLIB_HOME/plugins/Linux64_GCC-9.3/GenericTrace.so" \

```

```
-C TRACE.GenericTrace.trace-sources=WAYPOINT
```

### 1.13.2.2 Configuring data prefetching

The purpose of data prefetch modeling is to make the contents of the data cache more closely resemble those on a system with a hardware prefetcher. A default data prefetcher is supplied, which is relatively configurable. It is not intended to match any specific processor.

To run the model with data prefetch modeling enabled using the default data prefetcher and default parameters, use the following parameters:

```
-C cache_state_modelled=true --plugin "<<internal><DataPrefetch>>" -C  
cluster0.dcache-prefetch_enabled=1
```

When the model exits, it reports how many prefetches were issued and how many cache hits on recently prefetched data were detected. The performance impact is about 10% compared to running with cache state modeling enabled.

By default, a data prefetch plug-in attaches to all processors and clusters in a system and maintains independent internal state for each processor. To change this, for example to use a different number of tracked streams on big and LITTLE cores, load the plug-in twice and pass a different `.cluster` parameter to each instance:

```
--plugin "DP_BIG=<<internal><DataPrefetch>>" --plugin  
"DP_LITTLE=<<internal><DataPrefetch>>" \  
-C DataPrefetch.DP_BIG.cluster=0 -C DataPrefetch.DP_LITTLE.cluster=1 \  
-C DataPrefetch.DP_BIG.lfb_entries=16 -C DataPrefetch.DP_LITTLE.lfb_entries=4
```

The names `DP_BIG` and `DP_LITTLE` are examples. They can be any names you choose.

The example prefetcher is a basic stride-detecting prefetcher, but relatively configurable using the following parameters:

**Table 1-3: Parameters for the example prefetcher**

Parameter	Description
<code>history_length</code>	Length of history to maintain.
<code>history_threshold</code>	Misses allowed in history before prefetch.
<code>lfb_entries</code>	Number of access streams to track.
<code>mbs_expire</code>	Non-hitting loads allowed before the prefetcher stops tracking a potential access stream.
<code>pf_count</code>	Number of prefetch streams available.
<code>pf_tracker_count</code>	Number of prefetches tracked.
<code>pf_initial_number</code>	Initial number of prefetches for new stream.
<code>prefetch_all_levels</code>	Prefetch to all cache levels rather than just the lowest level.

## Usage

An access stream is created whenever a load is made to an address that is not within three cache lines of a previously observed load; this might overwrite a previously created access stream. When a consistent stride has been observed, that is when addresses  $N$ ,  $N+\text{delta}$ ,  $N+2*\text{delta}$  are seen, a prefetch stream is allocated with stride `delta` and a lifetime of `pf_initial_number`.

Prefetches are issued in a round-robin fashion from active prefetch streams (the lifetime goes down by one each time a prefetch is issued) whenever there have been fewer than `history_threshold` cache misses among the last `history_length` loads. The rationale is that if lots of cache hits are occurring, there should be available bandwidth on the memory interface to be used by prefetching.

Issued prefetches are tracked in a circular list of size `pf_tracker_count`, and if the prefetcher sees a load to an address in this circular list, it increments the lifetime of the prefetch stream that issued the successful prefetch.



Prefetches are to physical addresses; a prefetch stream expires when it reaches the end of a 4 KB region.

For more information, see:

- [Instruction and data prefetching](#)
- [Configuring instruction prefetching](#)

### 1.13.3 Cache and TLB latency

You can configure latency for different cache operations for Cortex®-A processor models by setting model parameters.

The following table shows examples of parameters for each cache operation:

**Table 1-4: Cache and TLB latency parameters**

Category	Example parameters
Read access latency for L1 D/I or L2 cache	<code>dcache-read_access_latency</code>
Separate latencies for read hits and misses. The total latency for a read access is the sum of the read access latency and the hit or miss latency.	<code>dcache-hit_latency</code> , <code>dcache-miss_latency</code>
Write access latency for L1 D-cache or L2 cache.	<code>dcache-write_access_latency</code>
Maintenance operation latency for L1 D-cache L1 I-cache, or L2 cache.	<code>dcache-maintenance_latency</code>
Latency for snoop accesses that perform a data transfer for L1 D-cache or L2 cache	<code>dcache-snoop_data_transfer_latency</code>
Latency for snoop accesses issued by L2 cache.	<code>l2cache-snoop_issue_latency</code>
TLB and page table walk latencies	<code>tlb_latency</code>



## Usage

- Enable cache state modeling before setting latency parameters, for example using `dcache-state_modelled` and `icache-state_modelled`.
- All latency values are in clock ticks.
- For reads and writes, you can specify latency per access, for example `dcache-read_access_latency`, or per byte, for example `dcache-read_latency`. If both are set, the per-access value takes precedence.

## 2. Protocols

Components communicate through connected ports. Ports have protocols that define the function calls for different connections.

### 2.1 AMBAPV protocol

Defined in `$PVLIB_HOME/LISA/AMBAPVProtocol.lisa`.

#### About AMBAPV protocol

The AMBAPV protocol defines behaviors for single read and single write transactions. This covers Arm AMBA AXI5, AXI4, AXI3, AHB, and APB bus protocol families, at the PV level.

In addition, the AMBAPV protocol supports AMBA protocol additional control information:

- Protection units.
- Exclusive access and locked access mechanisms.
- Atomic accesses, including exclusive accesses, locked accesses, and atomic transactions.
- System-level caching.
- Distributed Virtual Memory transactions (DVM).
- Memory Tagging Extension (MTE).
- Memory System Resource Partitioning and Monitoring (MPAM).
- Memory Encryption Contexts (MEC).

It always returns the original data at the target address.

The generic payload data is formatted as an array of bytes in order of ascending bus address. This means that irrespective of the host machine endianness or modeled bus width:

- A little endian master must write the bytes of a word in increasing significance as the array index increases.
- A big endian master must write the bytes of a word in decreasing significance as the array index increases.

A master or slave whose endianness does not match the endianness of the host machine must endian swap any access to the payload data that is wider than one byte. The same byte ordering rule applies to memory accesses using DMI pointers.

AMBAPV provides the following behaviors:

#### **atomic\_compare()**

```
optional slave behavior atomic_compare(int socket_id,  
                                       const sc_dt::uint64 & addr,  
                                       unsigned_char * data,
```

```

        unsigned int length,
        unsigned int size,
        amba_pv::amba_pv_control * ctrl,
        sc_core::sc_time & t) : amba_pv::amba_pv_resp_t;

```

This optional slave behavior completes an AtomicCompare transaction with the specified compare value and swap value. If the compare value equals the values at the given address, the swap value is written to the addressed location.

### **atomic\_load()**

```

optional slave behavior atomic_load(int socket_id,
        const sc_dt::uint64 & addr,
        unsigned char * data,
        unsigned int length,
        unsigned int size,
        amba_pv::amba_pv_control * ctrl,
        amba_pv::amba_pv_atomic_subop_t subop,
        amba_pv::amba_pv_atomic_endianness_t endianness,
        sc_core::sc_time & t) : amba_pv::amba_pv_resp_t;

```

This optional slave behavior complete an AtomicLoad transaction with the specified data. The data is used by the atomic transaction in the specified endianness.

### **atomic\_store()**

```

optional slave behavior atomic_store(int socket_id,
        const sc_dt::uint64 & addr,
        unsigned char * data,
        unsigned int length,
        unsigned int size,
        amba_pv::amba_pv_control * ctrl,
        amba_pv::amba_pv_atomic_subop_t subop,
        amba_pv::amba_pv_atomic_endianness_t endianness,
        sc_core::sc_time & t) : amba_pv::amba_pv_resp_t;

```

This optional slave behavior complete an AtomicStore transaction with the specified data. The data is used by the atomic transaction in the specified endianness.

### **atomic\_swap()**

```

optional slave behavior atomic_swap(int socket_id,
        const sc_dt::uint64 & addr,
        unsigned char * data,
        unsigned int length,
        unsigned int size,
        amba_pv::amba_pv_control * ctrl,
        sc_core::sc_time & t) : amba_pv::amba_pv_resp_t;

```

This optional slave behavior completes an AtomicSwap transaction with the specified data, which is written to the specified address. The original data is returned.

### **b\_transport()**

```

optional slave behavior b_transport(int socket_id,
        amba_pv::amba_pv_transaction & trans,
        sc_core::sc_time & t) : void;

```

This is an optional slave behavior for blocking transport. It completes a single transaction using the blocking transport interface. The `amba_pv::amba_pv_extension` must be added to the transaction before calling this behavior. The `socket_id` parameter must be set to 0 in this context.

### **debug\_read()**

```
optional slave behavior debug_read(int socket_id,
                                   const sc_dt::uint64 & addr,
                                   unsigned char * data,
                                   unsigned int length,
                                   amba_pv::amba_pv_control * ctrl) : unsigned int;
```

This optional slave behavior completes a debug read transaction from a given address without causing any side effects. Specify the number of bytes to read in the `length` parameter. The number of successfully read values is returned. Additional AMBA protocol control information can be specified in the `ctrl` parameter. The `socket_id` parameter must be set to 0 in this context. This behavior is empty by default and returns 0.

### **debug\_write()**

```
optional slave behavior debug_write(int socket_id,
                                    const sc_dt::uint64 & addr,
                                    unsigned char * data,
                                    unsigned int length,
                                    amba_pv::amba_pv_control * ctrl) : unsigned int;
```

This optional slave behavior completes a debug write transaction to a given address without causing any side effects. Specify the number of bytes to write in the `length` parameter. The number of successfully written values is returned. Additional AMBA protocol control information can be specified in the `ctrl` parameter. The `socket_id` parameter must be set to 0 in this context. This behavior is empty by default and returns 0.

### **get\_direct\_mem\_ptr()**

```
optional slave behavior get_direct_mem_ptr(int socket_id,
                                           amba_pv::amba_pv_transaction & trans,
                                           tlm::tlm_dmi & dmi_data) : bool;
```

This optional slave behavior requests a DMI access to a given address. It returns a reference to a DMI descriptor that contains the bounds of the granted DMI region. Returns `true` if a DMI region is granted, `false` otherwise.

This behaviour can also be used to request tag DMI by setting the corresponding MTE attributes. Tag DMI is granted independently of data DMI, and the tag DMI grant result and descriptor are returned via the MTE attributes.

### **invalidate\_direct\_mem\_ptr()**

```
optional master behavior invalidate_direct_mem_ptr(int socket_id,
                                                  sc_dt::uint64 start_range,
                                                  sc_dt::uint64 end_range) : void;
```

This optional master behavior invalidates a DMI request. It invalidates DMI pointers that were previously established for the given DMI region. The `socket_id` parameter is 0 in this context.

An invalidation applies to both data DMI and tag DMI. Any cached tag DMI whose covered tag granules overlap the invalidation range shall also be invalidated.

### **read()**

```
optional slave behavior read(int socket_id,
                             const sc_dt::uint64 & addr,
                             unsigned char * data,
                             unsigned int size,
                             amba_pv::amba_pv_control * ctrl,
                             sc_core::sc_time & t) : amba_pv::amba_pv_resp_t;
```

This optional slave behavior completes a single read transaction at the given address for the given size in bytes. Additional AMBA protocol control information can be specified using the `ctrl` parameter. The `socket_id` parameter must be set to 0 in this context.

### **transport\_dbg()**

```
optional slave behavior transport_dbg(int socket_id,
                                      amba_pv::amba_pv_transaction & trans) : unsigned int;
```

This optional slave behavior implements the TLM debug transport interface. An `amba_pv::amba_pv_extension` object must be added to the transaction before calling this behavior. The `socket_id` parameter must be set to 0 in this context.

### **write()**

```
optional slave behavior write(int socket_id,
                              const sc_dt::uint64 & addr,
                              unsigned char * data,
                              unsigned int size,
                              amba_pv::amba_pv_control * ctrl,
                              unsigned char * strb,
                              sc_core::sc_time & t) : amba_pv::amba_pv_resp_t;
```

This optional slave behavior completes a single write transaction at the given address with specified data and write strobes. The size of the data is specified in bytes. Additional AMBA protocol control information can be specified using the `ctrl` parameter. The `socket_id` parameter must be set to 0 in this context.

## **2.2 AMBAPVACE protocol**

Defined in `$PVLIB_HOME/LISA/AMBAPVACEProtocol.lisa`.

### **About AMBAPVACE protocol**

This protocol defines behaviors for bus transactions. This covers Arm AMBA ACE and DVM bus protocol families, all at the PV level.

In addition, this protocol provides support for AMBA protocol additional extension information:

- Secure and privileged accesses.
- Atomic accesses.
- System-level caching and buffering control.
- Cache coherency transactions (ACE-Lite).
- Bi-directional cache coherency transactions (ACE).
- Distributed Virtual Memory transactions (DVM).
- Memory Tagging Extension (MTE).
- Memory System Resource Partitioning and Monitoring (MPAM).
- Memory Encryption Contexts (MEC).

The generic payload data is in the format of an array of bytes in order of ascending bus address. This means that irrespective of the host machine endianness or modeled bus width:

- A little endian master must write the bytes of a word in increasing significance as the array index increases.
- A big endian master must write the bytes of a word in decreasing significance as the array index increases.

A master or slave whose endianness does not match the endianness of the host machine must endian swap any access to the payload data that is wider than one byte. The same byte ordering rule applies to memory accesses using DMI pointers.

### Special considerations for ACE and cache coherent interconnects

An ACE interconnect model must be able to cope with concurrent transactions in accordance with the hazard avoidance and prioritization rules in the ACE specification. Any external bus request, downstream transaction or upstream snoop transaction, can potentially cause a transaction to stall and the calling thread to be blocked, resulting in any number of other threads being scheduled.

To maintain memory coherency, apply these rules for debug transactions:

#### debug reads

The bus must return data that represents the values that the bus master expects to observe if it issues a bus read. This must not modify the state of any bus components.

#### debug writes

These must modify the contents of all copies of the location being accessed, so that a subsequent read from this location returns the data in the debug-write request. The debug write must not modify any other state, such as cache tags, clean/dirty/shared/unique MOESI state.

The implications for a coherent interconnect are that incoming debug transactions must be broadcast back upstream as debug snoop transactions to all ports other than the one the request came in on. Incoming debug snoops must propagate upwards. Debug reads can terminate as soon as they hit a cache. Debug writes must continue until they propagate to all possible copies of the location, including downstream to main memory.

For cases where a debug transaction hazards with non-debug transactions that are in-flight, the debug transaction must observe a weak memory-order model. Any component that can block a thread whilst responsible for the payload of an in-flight transaction must take particular care. In these cases, the debug transaction must be hazarded against the in-flight payload to ensure that debug reads do not return stale data and debug writes do not cause cache incoherency.

Only use DMI when you can guarantee that subsequent transactions do not result in any state transitions. This means, in general, do not use DMI for ACE coherent cacheable transactions.

AMBAPVACE provides the following behaviors:

### **b\_snoop()**

```
optional master behavior b_snoop(int socket_id,
                                amba_pv::amba_pv_transaction & trans,
                                sc_core::sc_time & t) : void;
```

This master behavior implements an upstream snooping TLM blocking transport interface. You must add an `amba_pv::amba_pv_extension` object to the transaction before calling this behavior, setting the `socket_id` parameter to 0 in this context.

### **b\_transport()**

```
slave behavior b_transport(int socket_id,
                           amba_pv::amba_pv_transaction & trans,
                           sc_core::sc_time & t) : void;
```

This slave behavior implements the TLM blocking transport interface. An `amba_pv::amba_pv_extension` object must be added to the transaction before calling this behavior. The `socket_id` parameter must be set to 0 in this context.

### **get\_direct\_mem\_ptr()**

```
optional slave behavior get_direct_mem_ptr(int socket_id,
                                           amba_pv::amba_pv_transaction & trans,
                                           tlm::tlm_dmi & dmi_data) : bool;
```

This optional slave behavior is for requesting a DMI access to a given address. It returns a reference to a DMI descriptor that contains the bounds of the granted DMI region. Returns true if a DMI region is granted, false otherwise. You must add an `amba_pv::amba_pv_extension` object to the transaction before calling this behavior, setting the `socket_id` parameter to 0 in this context.

This behaviour can also be used to request tag DMI by setting the corresponding MTE attributes. Tag DMI is granted independently of data DMI, and the tag DMI grant result and descriptor are returned via the MTE attributes.

### **invalidate\_direct\_mem\_ptr()**

```
optional master behavior invalidate_direct_mem_ptr(int socket_id,
                                                  sc_dt::uint64 start_range,
                                                  sc_dt::uint64 end_range) : void;
```

Use this optional master behavior to invalidate a DMI request. It invalidates DMI pointers that were previously established for the given DMI region. The `socket_id` parameter is 0 in this context.

An invalidation applies to both data DMI and tag DMI. Any cached tag DMI whose covered tag granules overlap the invalidation range shall also be invalidated.

### **snoop\_dbg()**

```
optional master behavior snoop_dbg(int socket_id,
                                   amba_pv::amba_pv_transaction & trans) : unsigned int;
```

This optional master behavior implements an upstream snooping TLM debug transport interface. You must add an `amba_pv::amba_pv_extension` object to the transaction before calling this behavior, setting the `socket_id` parameter to 0 in this context.

### **transport\_dbg()**

```
optional slave behavior transport_dbg(int socket_id,
                                      amba_pv::amba_pv_transaction & trans) : unsigned int;
```

This optional slave behavior implements the TLM debug transport interface. You must add an `amba_pv::amba_pv_extension` object to the transaction before calling this behavior, setting the `socket_id` parameter to 0 in this context.

## 2.3 AMBAPVSignal protocol

Defined in `$PVLIB_HOME/LISA/AMBAPVSignalProtocol.lisa`.

### About AMBAPVSignal protocol

This protocol defines a single behavior to permit masters to change the state of signals such as interrupts. AMBA3 does not cover this behavior, but the AMBA-PV components do provide it.

AMBAPVSignal provides the following behaviors:

### **set\_state()**

```
slave behavior set_state(int export_id,
                         const_bool & state) : void;
```

Transfers a signal state. The `export_id` parameter must be set to 0 in this context.



## 2.4 AMBAPVSignalState protocol

Defined in `$PVLIB_HOME/LISA/AMBAPVSignalProtocol.lisa`.

### About AMBAPVSignalState protocol

This protocol defines two behaviors that permit a master to change the state of signals such as interrupts and to retrieve the state of such signals from slaves. This behavior is not covered by AMBA3, but is provided with the AMBA-PV components.

AMBAPVSignalState provides the following behaviors:

#### **get\_state()**

```
slave behavior get_state(int export_id,  
                        tlm::tlm_tag<bool> * t) : bool;
```

Retrieves a signal state. The `export_id` parameter must be set to 0, and the `t` parameter must be set to `NULL`, in this context.

#### **set\_state()**

```
slave behavior set_state(int export_id,  
                        const bool & state) : void;
```

Transfers a signal state. The `export_id` parameter must be set to 0 in this context.

## 2.5 AMBAPVValue protocol

Defined in `$PVLIB_HOME/LISA/AMBAPVValueProtocol.lisa`.

### About AMBAPVValue protocol

This protocol models propagation of 32-bit integer values between components.

Strictly speaking this behavior is not covered by AMBA3, but is provided with the AMBA-PV components.

AMBAPVValue provides the following behaviors:

#### **set\_state()**

```
slave behavior set_state(int export_id,  
                        const uint32_t & value) : void;
```

Transfers a value. The `export_id` parameter must be set to 0 in this context.

## 2.6 AMBAPVValue64 protocol

Defined in `$PVLIB_HOME/LISA/AMBAPVValue64Protocol.lisa`.

### About AMBAPVValue64 protocol

This protocol models propagation of 64-bit integer values between components.

Strictly speaking this behavior is not covered by AMBA3, but is provided with the AMBA-PV components.

AMBAPVValue64 provides the following behaviors:

#### **set\_state()**

```
slave behavior set_state(int export_id,
                        const uint64_t & value) : void;
```

Transfers a value. The `export_id` parameter must be set to 0 in this context.

## 2.7 AMBAPVValueState protocol

Defined in `$PVLIB_HOME/LISA/AMBAPVValueProtocol.lisa`.

### About AMBAPVValueState protocol

This protocol permits propagation of 32-bit integer values between components and their retrieval from slaves.

Strictly speaking this behavior is not covered by AMBA3, but is provided with the AMBA-PV components.

AMBAPVValueState provides the following behaviors:

#### **get\_state()**

```
slave behavior get_state(int export_id,
                        tlm::tlm_tag<uint32_t> * t) : uint32_t;
```

Retrieves a value. The `export_id` parameter must be set to 0 and the `t` parameter must be set to `NULL`, in this context.

#### **set\_state()**

```
slave behavior set_state(int export_id,
                        const uint32_t & value) : void;
```

Transfers a value. The `export_id` parameter must be set to 0 in this context.

## 2.8 AMBAPVValueState64 protocol

Defined in `$PVLIB_HOME/LISA/AMBAPVValue64Protocol.lisa`.

### About AMBAPVValueState64 protocol

This protocol permits propagation of 64-bit integer values between components and their retrieval from slaves.

Strictly speaking this behavior is not covered by AMBA3, but is provided with the AMBA-PV components.

AMBAPVValueState64 provides the following behaviors:

#### **get\_state()**

```
slave behavior get_state(int export_id,  
                        tlm::tlm_tag<uint64_t> * t) : uint64_t;
```

Retrieves a value. The `export_id` parameter must be set to 0, and the `t` parameter must be set to `NULL`, in this context.

#### **set\_state()**

```
slave behavior set_state(int export_id,  
                        const uint64_t & value) : void;
```

Transfers a value. The `export_id` parameter must be set to 0 in this context.

## 2.9 AsyncSignalCallback protocol

Defined in `$PVLIB_HOME/LISA/AsyncSignalCallbackProtocol.lisa`.

### About AsyncSignalCallback protocol

This protocol is used to get callbacks from the AsyncSignal component. The component usually implements internal slave ports of this protocol and connects them to the `AsyncSignal.async_callback` master port.

AsyncSignalCallback provides the following behaviors:

#### **signal()**

```
slave behavior signal() : void;
```

Called from the AsyncSignal component. Only ever called on the simulation thread.

This is called asynchronously after a non-simulation thread has called `AsyncSignal.async_control.signal()`.

## 2.10 AsyncSignalControl protocol

Defined in `$PVLIB_HOME/LISA/AsyncSignalControlProtocol.lisa`.

### About AsyncSignalControl protocol

Non-simulation threads use this protocol to cause events on the simulation thread through the AsyncSignal component.

AsyncSignalControl provides the following behaviors:

#### **signal()**

```
slave behavior signal() : void;
```

Cause event on the simulation thread. Usually called by non-simulation threads.

Calling this behavior causes the `AsyncSignal.async_callback.signal()` function to be called asynchronously later. It is safe to call this function from any thread.

## 2.11 AudioControl protocol

Defined in `$PVLIB_HOME/LISA/AudioControlProtocol.lisa`.

### About AudioControl protocol

This protocol has get and release audio buffer behaviors.

AudioControl provides the following behaviors:

#### **getPVAudioBuffer()**

```
slave behavior getPVAudioBuffer(uint32_t depth) : PVAudioBuffer*;
```

Get an underlying host buffer for audio output.

#### **releasePVAudioBuffer()**

```
slave behavior releasePVAudioBuffer(PVAudioBuffer* buf) : void;
```

Release an underlying host buffer.

## 2.12 CADIDisassemblerProtocol protocol

Defined in `$PVLIB_HOME/LISA/CADIDisassemblerProtocol.lisa`.

### About CADIDisassemblerProtocol protocol

To support disassembly, implement all of these functions. None of them is optional.

These functions are in a different port, of type `CADIDisassemblerProtocol`. They can have any name and only need to be implemented to expose disassembly in the debugger. The functionality of this port is then exposed by `CADIProtocol::CADIGetDisassembler()`.

See [CADIProtocol protocol](#) for information on how to use this port and `CADIDisassemblerAdapter`.

`CADIDisassemblerProtocol` provides the following behaviors:

#### GetAddressForSourceReference ()

```
slave behavior GetAddressForSourceReference(const char *sourceFile, uint32_t
sourceLine, eslapi::CADIAddr_t &address) : eslapi::CADIDisassemblerStatus;
```

Get the first address for the given source line and file.

#### GetCurrentMode ()

```
slave behavior GetCurrentMode() : uint32_t;
```

Get the most suitable mode of disassembly, based on the current state of the variables of the component.

If modes are not supported by this target, return 0. If modes are supported, return  $0 < x \leq \text{GetModeCount}()$ .

#### GetDisassembly ()

```
slave behavior GetDisassembly(eslapi::CADIDisassemblerCB *callback_,
                             const eslapi::CADIAddr_t &address,
                             eslapi::CADIAddr_t &nextAddr,
                             const uint32_t mode,
                             uint32_t desiredCount) :
    eslapi::CADIDisassemblerStatus;
```

The main disassembler function for standard type disassembly.

The component must call `callback_` for all disassembler lines for the specified `address` and `desiredCount`, and must finally set `nextAddr` to the next disassembled address at that point after the requested block.

**GetInstructionType()**

```
slave behavior GetInstructionType(const eslapi::CADIAddr_t
    &address, eslapi::CADIDisassemblerInstructionType &insn_type) :
    eslapi::CADIDisassemblerStatus;
```

Query if an instruction is a call instruction.

Components must set `insn_type = eslapi::CADI_DISASSEMBLER_INSTRUCTION_TYPE_NOCALL` and return `eslapi::CADI_DISASSEMBLER_STATUS_OK`.

**GetModeCount()**

```
slave behavior GetModeCount() : uint32_t;
```

Return the number of supported disassembler modes. At least one mode must be returned.

**GetModeNames()**

```
slave behavior GetModeNames(eslapi::CADIDisassemblerCB *callback_) : void;
```

Query the names of all supported modes.

Triggers callbacks to `CADIDisassemblerCB::ReceiveModeName()`, once for every mode. A component that only supports one mode calls, for example, `callback_ -> ReceiveModeName(0, "Normal")`; only once. This is similar for multiple modes with different names and IDs.

**GetSourceReferenceForAddress()**

```
slave behavior GetSourceReferenceForAddress(eslapi::CADIDisassemblerCB *callback_,
    const eslapi::CADIAddr_t &address) : eslapi::CADIDisassemblerStatus;
```

Retrieves source-level information. Triggers a call of `CADIDisassemblerCB::ReceiveSourceReference()`.

**GetType()**

```
slave behavior GetType() : eslapi::CADIDisassemblerType;
```

Distinguish between different types of disassembly. Components must always return `eslapi::CADI_DISASSEMBLER_TYPE_STANDARD`.

## 2.13 CADIProtocol protocol

Defined in `$PVLIB_HOME/LISA/CADIProtocol.lisa`.

**About CADIProtocol protocol**

This protocol supports debugging.

By default, LISA components do not support breakpoints. To add breakpoint support:

- Define an internal slave port of this type, whose name must be `cadi_port`
- Implement all of the following functions:

```
optional slave behavior CADIBptGetList(uint32_t, uint32_t, uint32_t *,
    eslapi::CADIBptDescription_t *) : eslapi::CADIReturn_t;
optional slave behavior CADIBptRead(eslapi::CADIBptNumber_t,
    eslapi::CADIBptRequest_t *) : eslapi::CADIReturn_t;
optional slave behavior CADIBptSet(eslapi::CADIBptRequest_t *,
    eslapi::CADIBptNumber_t *) : eslapi::CADIReturn_t;
optional slave behavior
    CADIBptClear(eslapi::CADIBptNumber_t) : eslapi::CADIReturn_t;
optional slave behavior CADIBptConfigure(eslapi::CADIBptNumber_t,
    eslapi::CADIBptConfigure_t) : eslapi::CADIReturn_t;
optional slave behavior CADIModifyTargetFeatures(eslapi::CADITargetFeatures_t
    *) : eslapi::CADIReturn_t;
```

In addition to implementing these functions, when an enabled breakpoint is hit, the component must:

- Call `simBreakpointHit(bptNumber)` for each breakpoint that was hit (one or more, usually just one).
- Call `simHalt()` once, after all `simBreakpointHit()` calls. The `simHalt()` call must be the last call in the sequence.

CADIProtocol provides the following behaviors:

### **CADIBptClear()**

```
optional slave behavior CADIBptClear(eslapi::CADIBptNumber_t) :
    eslapi::CADIReturn_t;
```

Clear the breakpoint specified by `CADIBptNumber_t`.

### **CADIBptConfigure()**

```
optional slave behavior CADIBptConfigure(eslapi::CADIBptNumber_t,
    eslapi::CADIBptConfigure_t) : eslapi::CADIReturn_t;
```

Re-configure an existing breakpoint.

### **CADIBptGetList()**

```
optional slave behavior CADIBptGetList(uint32_t, uint32_t, uint32_t *,
    eslapi::CADIBptDescription_t *) : eslapi::CADIReturn_t;
```

Provides a list of current breakpoints. The component must maintain and keep track of all existing breakpoints.

### **CADIBptRead()**

```
optional slave behavior CADIBptRead(eslapi::CADIBptNumber_t,
    eslapi::CADIBptRequest_t *) : eslapi::CADIReturn_t;
```

Provides a `CADIBptRequest_t` object for the breakpoint with number `CADIBptNumber_t`

### **CADIBptSet()**

```
optional slave behavior CADIBptSet(eslapi::CADIBptRequest_t *,
    eslapi::CADIBptNumber_t *) : eslapi::CADIReturn_t;
```

Create a new breakpoint. The breakpoint number is returned.

### **CADIExecSingleStep()**

```
optional slave behavior CADIExecSingleStep(uint32_t instructionCount, int8_t
    stepCycle, int8_t stepOver) : eslapi::CADIReturn_t;
```

Single stepping needs support from the individual model. Run and stop are always handled globally. This behavior implements instruction stepping. It must set up an internal state that stops the simulation when the requested number of instructions is executed completely, exactly like a breakpoint. It must call `simRun()` from within `CADIExecSingleStep()` after setting up this stepping state, and later it must call `simHalt()` when the execution of the required number of instructions finishes.

### **CADIGetCycleCount()**

```
optional slave behavior CADIGetCycleCount(uint64_t &instructionCount, bool
    systemCycles) : eslapi::CADIReturn_t;
```

Get cycle count. By implementing this function, the component can enable the cycle count display.



Note

Fast Models systems are not cycle accurate, so you usually only implement an instruction counter, if at all.

### **CADIGetDisassembler()**

```
optional slave behavior CADIGetDisassembler() : eslapi::CADIDisassembler*;
```

To provide disassembly, a component must implement the `CADIGetDisassembler()` behavior and return a `CADIDisassembler` interface implementation. This automatically follows the `CADI::CADIGetDisassembler()` and the `CADI::ObtainInterface("eslapi.CADIDisassembler2")` functions.

To do this, instantiate a `CADIDisassemblerAdapter` object in behavior `init()` and return its address in the `CADIGetDisassembler()` function. This object must point to an internal slave port that implements the `CADIDisassemblerProtocol` protocol.

Skeleton code for implementing disassembly:

```
component FOO
{
```



```

behavior init()
{
    disassemblerAdapter = new
    CADIDisassemblerAdapter(disassPort.getAbstractInterface());
    // ...
}
internal slave port <CADIProtocol> cadi_port
{
    slave behavior CADIGetDisassembler():eslapi::CADIDisassembler*
    {
        return disassemblerAdapter;
    }
    // ...
}
internal slave port<CADIDisassemblerProtocol> disassPort
{
    // ...
}
}

```

### **CADIGetInstructionCount()**

```

optional slave behavior CADIGetInstructionCount(uint64_t &instructionCount) :
    eslapi::CADIReturn_t;

```

Get the instruction count. By implementing this function, the component can enable the instruction count display.

### **CADIModifyTargetFeatures()**

```

optional slave behavior CADIModifyTargetFeatures(eslapi::CADITargetFeatures_t *) :
    eslapi::CADIReturn_t;

```

Allows you to override the default `CADITargetFeatures_t` that System Generator provides for this component just before it is returned to the debugger.

Note that this method is not part of the CADI specification.

Specifically, a component that wants to support any kind of breakpoint must override the `handledBreakpoints` and `nrBreakpointsAvailable` fields of `CADITargetFeatures_t`.

For example, to support virtually infinite code and register breakpoints:

```

targetFeatures->handledBreakpoints = CADI_TARGET_FEATURE_BPT_PROGRAM |
    CADI_TARGET_FEATURE_BPT_REGISTER;

targetFeatures->nrBreakpointsAvailable = 0x7fffffff;

```

### **callbackModeChange()**

```

optional slave behavior callbackModeChange(uint32_t newMode, eslapi::CADIBptNumber_t
    bptNumber) : void;

```

Forwards all `modeChange()` callbacks to the target component. The target should generally ignore all of these except when implementing `CADIExecSingleStep()`.

This function is for debugging purposes only. Do not implement it. The function must not alter the state of any component in any way.

## 2.14 CCI500\_AddressDecoderProtocol protocol

Defined in `$PVLIB_HOME/LISA/CCI500_AddressDecoderProtocol.lisa`.

### About CCI500\_AddressDecoderProtocol

CCI-5x0 allows a customer-supplied decode policy to route transactions across the downstream ports of the interconnect. This can be done by connecting a valid `address_decoder` to the `address_decoder` port on the CCI-5x0.

This protocol is used for `address_decoder` ports in both CCI500 and CCI550.

`decode_by_4KiB_addr` is the main behavior that needs to be implemented which the CCI-5x0 model uses to figure out the downstream port that a transaction needs to be routed to.

If you wish to abort a transaction then return `PVBUSMAPPER_ABORT` to `decode_by_4KiB_addr`.

### Limitations

- In the RTL, the customer can stripe across several ports at a granularity less than 4 KiB in order to load balance across a memory controller's ports (or multiple memory controllers).
- In the model, we do not support sub-4KiB decode/stripping. This is not anticipated to be a problem as there is little point in stripping to modelled memory controllers.
- In the RTL, it is a requirement that the decode be static after reset. However, in the model then you can change it any time up until the first transaction, after that point then you must keep the decode static until the next reset.
- The decode *may* depend on `upstream_port_index` only to the extent that a particular `upstream_port_index` might not be allowed to communicate with a particular downstream port. However, having an address map that depends on the `upstream_port_index` will mostly likely produce coherency issues.

`CCI500_AddressDecoderProtocol` provides the following behaviors:

### `configuration()`

```
optional slave behavior configuration(const CCI5x0_AddressDecoderConfiguration&) :
void;
```

If the slave implements this then the configuration will be told to the address decoder at reset time.

This is useful if the configuration of the system can be changed at init time and can avoid replicating the parameters from the CCI-5x0 to the decoder.

### `decode_by_4KiB_addr()`

```
slave behavior decode_by_4KiB_addr( unsigned upstream_port_index,
```

```
bool    is_read_or_cmo_,
uint64_t address_,
bool    ns_ ) : unsigned;
```



The implementation of this must not cause any thread switch during its execution.

### get\_squash\_record()

```
optional slave behavior get_squash_record( unsigned downstream_port_index_,
                                           unsigned* out_lsb_bitpos_,
                                           unsigned*
                                           out_number_of_bits_to_squash_ ) : void;
```

This is used to get the output address transformation to make on the specified `downstream_port_index_`. The implementation will read all the squash records for each of the downstream ports the first time it receives a transaction.

```
addr[*out_lsb_bitpos_ + out_number_of_bits_to_squash_ : *out_lsb_bitpos_]
```

will be sliced out of any address going to the specified port.

If you do not wish to perform any slicing, return 0 for `out_number_of_bits_to_squash_`

### reset()

```
optional slave behavior reset() : void;
```

This is called when CCI-500 is reset.

## 2.15 CCIInterconnectControl protocol

Defined in `$PVLIB_HOME/LISA/PVCache.lisa`.

### About CCIInterconnectControl

Internal only. Used only for the CCI400 model. The CCIRegisters component uses it to grab the control interface from the CCIInterconnect component. The control interface allows us to read/write various configuration options that determine where snoops should be sent etc.

CCIInterconnectControl provides the following behaviors:

### getControlIf()

```
slave behavior getControlIf() : CCIInterconnect::control_if*;
```

Get CCIInterconnect control\_if pointer.

## 2.16 ClockRateControl protocol

Defined in `$PVLIB_HOME/LISA/ClockRateControlProtocol.lisa`.

### About ClockRateControl protocol

Allow systems to dynamically modify the multiply/divide ratio of a ClockDivider component.

If a ClockDivider's ratio is changed, the frequency of its `clk_out` signal is immediately recalculated, along with any clocks derived from that signal.

Any active ClockTimers will automatically compute the number of ticks elapsed so far at the old clock rate, and continue counting down at the new rate. This may introduce a slight rounding error of a fraction of a tick.

ClockRateControl provides the following behaviors:

#### set()

```
peer behavior set(uint32_t mul, uint32_t div) : void;
```

Set clock rate using 32-bit values. The new clock rate =  $\text{mul} / \text{div}$ .

#### set64()

```
peer behavior set64(uint64_t mul, uint64_t div) : void;
```

Set clock rate using 64-bit values. The new clock rate =  $\text{mul} / \text{div}$ .

## 2.17 ClockSignal protocol

Defined in `$PVLIB_HOME/LISA/ClockSignalProtocol.lisa`.

### About ClockSignal protocol

A ClockSignal port represents a timebase of a given frequency. This is an opaque port type. It contains no user-accessible behavior.

ClockSignal output ports are provided on the following library components:

#### MasterClock

Produces a clock signal at a base clock rate, which can nominally be considered to be 1Hz.

#### ClockDivider

Can be used to take an input ClockSignal from a MasterClock or from another ClockDivider and generate an output that is related to the input signal by a given ratio.

ClockSignals can be used as input to CpuComponents, to define the core clock rate. They can also be used to drive the clock port of a `clockTimer` component, which can be used to generate events in the scheduler.



A ClockSignal does not actually define a fixed square-wave signal. It merely defines a frequency that can be used by counter timers.

Here is an example system using ClockSignals:

```
composition {
  masterclock : MasterClock;
  div_24MHz : ClockDivider(div = 1, mul = 24000000);
  timer : ClockTimer;
}
master port<TimerControl> timer_control;
slave port<TimerCallback> timer_callback {
  behavior signal() : uint32_t {
    // handle timed event here
    // ...
    return 10; // reschedule in 10 ticks of input clock.
  }
}
behavior start_timer() {
  timer_control.set(10); // start timer counting 10 ticks.
}
connection {
  masterclock.clk_out  => div_24MHz.clk_in;
  div_24MHz.clk_out    => timer.clk_in;
  self.timer_control  => timer.timer_control;
  timer.timer_callback => self.timer_callback;
}
```

clocksignal provides the following behaviors:

### **currentTicks()**

```
peer behavior currentTicks() : uint64_t;
```

Private internal method used between Scheduler components.

### **getClock()**

```
peer behavior getClock() : sg::FrequencySource*;
```

Private internal method used between Scheduler components.

### **rateInHz()**

```
peer behavior rateInHz() : double;
```

Private internal method used between Scheduler components.

**setClock()**

```
peer behavior setClock(sg::FrequencySource*) : void;
```

Private internal method used between Scheduler components.

## 2.18 CompoundPortLisa protocol

Defined in `$PVLIB_HOME/LISA/CompoundPort.lisa`.

CompoundPortLisa provides the following behaviors:

**connectFromExternalSlavePort()**

```
slave behavior connectFromExternalSlavePort      (const std::string & name,  
                                                  sg::Port *) : void;
```

**connectToExternalMasterPort()**

```
slave behavior connectToExternalMasterPort      (const std::string & name,  
                                                  sg::Port *) : void;
```

**disconnectFromExternalSlavePort()**

```
slave behavior disconnectFromExternalSlavePort  (const std::string & name,  
                                                  sg::Port *) : void;
```

**disconnectToExternalMasterPort()**

```
slave behavior disconnectToExternalMasterPort  (const std::string & name,  
                                                  sg::Port *) : void;
```

## 2.19 CoprocBusProtocol protocol

Defined in `$PVLIB_HOME/LISA/CoprocBusProtocol.lisa`.

**About CoprocBusProtocol protocol**

This protocol connects a coprocessor implementation with a CPU component, for instance ARMCortexM33CT.

A coprocessor must derive from the coprocessor callback interface, `Coprocessor`. It can implement the CDP, MCR, MRC, STC, LDC, MCRR, and MRRC instructions.

A coprocessor must be registered with a specific coprocessor number, by calling the `addCoprocessor()` method. You can only register an external coprocessor that is not already present in the CPU. If no coprocessor has been registered with the coprocessor number encoded in an instruction, the CPU raises a NOCP fault.

To register coprocessor instruction implementations with the CPU, you must initialize the function pointers. For example, the following code passes the function pointers to the Coprocessor constructor. This code was taken from the `$PVLIB_HOME/examples/LISA/FVP_Coproc_Demo/` example.

## Registering a coprocessor

```
...
class TestValCoprocessor : public Coprocessor
{
public:
    protocol_CoprocBusProtocol * coproc_bus;
    uint32_t coproc_number;
    uint32_t cp_reg[2][NUM_CP_REG] = {{0}}; // [0][NUM_CP_REG] --> Secure, [1]
[ NUM_CP_REG ] --> Non-Secure
    TestValCoprocessor()
        : Coprocessor(this, test_CDP, nullptr, test_MCR, nullptr, test_MRC, nullptr,
test_LDC, nullptr, test_STC, nullptr, test_MCRR, nullptr, test_MRRC, nullptr)
        , coproc_bus(nullptr)
        , coproc_number(0)
    {

...
    };

    PARAMETER { description("coprocessors number"), type(uint32_t), default(0x2),
min(0x0), max(16) } coprocessor_number; // CP num
    TestValCoprocessor test_cp;
}

behaviour init
{
...
    if (coproc_bus.addCoprocessor.implemented())
    {
        coproc_bus.addCoprocessor(&test_cp, coprocessor_number);
    }
}
```

## Coprocessor callback functions

A coprocessor can implement callback functions with these signatures.

Each function returns a `CoprocState` value to indicate the new transaction state of the coprocessor.

### CDP()

Perform a coprocessor data processing operation.

```
CoprocState CDP(void* context, uint32_t inst)
```

Parameters:

#### context

Context that was registered with the coprocessor interface.

#### inst

The coprocessor instruction being executed.

### MCR()

Perform a move to coprocessor register operation.

```
CoprocState MCR(void* context, uint32_t inst, uint32_t data)
```

Parameters:

**context**

Context that was registered with the coprocessor interface.

**inst**

The coprocessor instruction being executed.

**data**

Register contents.

**MRC ()**

Perform a move from coprocessor register operation.

```
CoprocState MRC(void* context, uint32_t inst, uint32_t* data)
```

Parameters:

**context**

Context that was registered with the coprocessor interface.

**inst**

The coprocessor instruction being executed.

**data**

Pointer to word to fill with coprocessor register contents.

**LDC ()**

Perform a load coprocessor register from memory operation.

```
CoprocState LDC(void* context, uint32_t inst, uint32_t data, CoprocState state)
```

Parameters:

**context**

Context that was registered with the coprocessor interface.

**inst**

The coprocessor instruction being executed.

**data**

Contents of current memory location to load into register.

**state**

Current state in a sequence of transactions.

**STC ()**

Perform a store coprocessor register to memory operation.

```
CoprocState STC(void* context, uint32_t inst, uint32_t data, CoprocState state)
```

Parameters:

**context**

Context that was registered with the coprocessor interface.



- inst**  
The coprocessor instruction being executed.
- data**  
Pointer to word to fill with coprocessor register contents to be transferred to memory.
- state**  
Current state in a sequence of transactions.

**MCRR ()**  
Perform a move to two coprocessor registers operation.

```
CoprocState MCRR(void* context, uint32_t inst, uint32_t data1, uint32_t data2)
```

Parameters:

**context**  
Context that was registered with the coprocessor interface.

**inst**  
The coprocessor instruction being executed.

**data1**  
First data word to load to a coprocessor register.

**data2**  
Second data word to load to a coprocessor register.

**MRRC ()**  
Perform a move from two coprocessor registers operation.

```
CoprocState MRRC(void* context, uint32_t inst, uint32_t* data1, uint32_t* data2)
```

Parameters:

**context**  
Context that was registered with the coprocessor interface.

**inst**  
The coprocessor instruction being executed.

**data1**  
Pointer to first word to fill with coprocessor register contents.

**data2**  
Pointer to second word to fill with coprocessor register contents.

**CoprocState values**

A CoprocState enum value is returned by coprocessor callback functions to indicate the new transaction state of the coprocessor. It is also used as a parameter for LDC and STC callback functions.

Value	State label	Description
0	CoprocOk	Complete/Ok.

Value	State label	Description
1	CoprocUndef	Undefined operation.
2	CoprocAbort	Data abort.
4	CoprocFirst	A parameter value for LDC and STC callback functions to indicate that this is the first data transfer in a sequence.
5	CoprocNext	A parameter value for LDC and STC callback functions to indicate that this is a subsequent data transfer in a sequence.
12	CoprocNop	Treat as a <b>NOP</b> .

CoprocBusProtocol provides the following behaviors:

#### **accessIsNonSecure()**

```
peer behavior accessIsNonSecure(void) : bool;
```

Checks the security state of the CPU, either true for non-secure, or false for secure.

#### **accessIsPriv()**

```
peer behavior accessIsPriv(void) : bool;
```

Checks whether the CPU state is privileged (true) or unprivileged (false).

#### **addCoprocesor()**

```
peer behavior addCoprocesor(Coprocessor*, int num) : void;
```

Registers the coprocessor with the CPU. `num` identifies which coprocessor to register it as.

#### **removeCoprocesor()**

```
peer behavior removeCoprocesor(Coprocessor*, int num) : void;
```

Unregisters the coprocessor from the CPU.

## 2.20 CounterInterface protocol

Defined in `$PVLIB_HOME/LISA/CounterInterface.lisa`.

### About CounterInterface protocol

Counter Interface protocol for communicating between counter interfaces and SoC-level memory mapped counter implementations.

CounterInterface provides the following behaviors:

#### **eventUpdate()**

```
slave behavior eventUpdate() : void;
```

Callback into event clients. May be called at any time. It is the client's responsibility to interrogate the physical counter to determine if its event should have been fired.

### **getCounterValue()**

```
master behavior getCounterValue() : uint64_t;
```

Get the absolute value of the physical timer.

### **requestEventUpdate()**

```
master behavior requestEventUpdate(uint64_t at) : void;
```

Request an eventUpdate at a particular time.

### **requestSignalUpdate()**

```
master behavior requestSignalUpdate(uint64_t at) : void;
```

Request a signalUpdate at a particular time.

### **setEnabled()**

```
slave behavior setEnabled(bool _bool_0) : void;
```

Communicate to the client if the counter module is enabled or not.

### **signalUpdate()**

```
slave behavior signalUpdate() : void;
```

Callback into counter clients. May be called at any time. It is the client's responsibility to interrogate the physical counter to determine if its timers should be signaled in response to the callback.

## 2.21 DVMMMessage protocol

Defined in `$PVLIB_HOME/LISA/DVMProtocol.lisa`.

DVMMMessage provides the following behaviors:

### **send()**

```
slave behavior send(DVM::Message*) : DVM::error_response_t;
```

Send DVM message.

## 2.22 EventBus protocol

Defined in `$PVLIB_HOME/LISA/EventBus.lisa`.

EventBus provides the following behaviors:

### **publishEventSource()**

```
peer behavior publishEventSource(uint32_t index, sg::EventSourceBase *src) : void;
```

## 2.23 Feature protocol

Defined in `$PVLIB_HOME/LISA/Feature.lisa`.

Feature provides the following behaviors:

### **setFeature()**

```
slave behavior setFeature(const char *featureName, const char *valueStr) : bool;
```

Set feature `featureName` to `valueStr`.

The encoding of `valueStr` is specific for each feature, but:

- For boolean features, use 0/1.
- For integer features, accept any base in C syntax, for example: 256, 0x100, 0400.

Returns true on success, false on error or if the feature is not supported.

## 2.24 FlashLoaderPort protocol

Defined in `$PVLIB_HOME/LISA/FlashLoaderPort.lisa`.

### **About FlashLoaderPort**

This protocol initializes the flash contents at model startup and saves flash contents to a file when the model terminates.

FlashLoaderPort provides the following behaviors:

### **loadFlashFile()**

```
slave behavior loadFlashFile(flash_loader::FlashLoader *loader) : uint32_t;
```

Initiate loading of the flash contents.

**saveFlashFile()**

```
slave behavior saveFlashFile(flash_loader::FlashLoader *loader) : uint32_t;
```

Save the flash contents to a file.

## 2.25 FrameTracingProtocol protocol

Defined in `$PVLIB_HOME/LISA/FrameTracingProtocol.lisa`.

**About FrameTracingProtocol protocol**

Port type used to connect to a `FrameTracingComponent`.

`FrameTracingProtocol` provides the following behaviors:

**beginFrame()**

```
slave behavior beginFrame(uint32_t width, uint32_t height, uint32_t bpp) : uint8_t*;
```

Request a memory buffer from the downstream `FrameTracingComponent` to write a frame. The returned buffer should be large enough to store `width * height` pixels of `bpp` bits each. The minimal expected returned buffer size is `width * height * ((bpp + 7) / 8)` bytes. A call to this method should be followed by a call to `endFrame()`.



Note

In case of an error, for example repeatedly calling `beginFrame()` before the matching `endFrame()`, this method returns a null pointer. If so, no data should be written there and there is no need to call `endFrame()`.

**endFrame()**

```
slave behavior endFrame() : void;
```

Notify the downstream `FrameTracingComponent` that the buffer provided by `beginFrame()` is now blitted with a frame and can be processed.

## 2.26 GICv3Comms protocol

Defined in `$PVLIB_HOME/LISA/GICv3Comms.lisa`.

**About GICv3Comms protocol**

Link for internal communications between GICv3 components.

The master is towards the top level, the slave is towards the CPU interface.

GICv3Comms provides the following behaviors:

### **sendTowardsCPU()**

```
slave behavior sendTowardsCPU(uint8_t len, const uint8_t* data) : void;
```

Sends byte stream towards the core.

### **sendTowardsTopLevel()**

```
master behavior sendTowardsTopLevel(uint8_t len, const uint8_t* data) : void;
```

Sends byte stream from the core.

### **setAXIManagerID()**

```
master behavior setAXIManagerID(uint64_t manager_id) : void;
```

Sets the ManagerID associated with the stream from the core.

## 2.27 GUIPollCallback protocol

Defined in \$PVLIB\_HOME/LISA/GUIPollCallbackPort.lisa.

### About GUIPollCallback protocol

Callback signal generated by a GUIPoll component. It allows a Visualisation component to continue to poll the GUI's event queue while the simulation is paused.

See the [GUIPoll](#) component for advice about using this protocol.

GUIPollCallback provides the following behaviors:

### **gui\_callback()**

```
slave behavior gui_callback() : void;
```

Client callback invocation, called at a period configured by the GUIPoll component.



This callback should only be used for updating a visualisation GUI. It should never be used for simulation events. See [GUIPoll](#) for more information.

---

## 2.28 ICS307Configuration protocol

Defined in `$PVLIB_HOME/LISA/ICS307ConfigurationPort.lisa`.

### About ICS307Configuration protocol

This protocol sets the divider ratio of an ICS307 component at runtime. The output clock rate alters accordingly and any dependent components react to the clock rate change according to their defined behavior.

ICS307Configuration provides the following behaviors:

#### setConfiguration()

```
peer behavior setConfiguration(uint32_t vdw, uint32_t rdw, uint32_t od) : void;
```

Set the parameters for deriving the clock divider ratio.

**vdw**

Range: 0-255.

**rdw**

Range: 0-255.

**od**

Range: 0-7.

## 2.29 InstructionCount protocol

Defined in `$PVLIB_HOME/LISA/InstructionCountProtocol.lisa`.

InstructionCount provides the following behaviors:

#### getRunState()

```
master behavior getRunState() : uint32_t;
```

Obtain the power/run status of the processor.

Value	State label	Description
0	UNKNOWN	Run status unknown, that is, simulation has not started
1	RUNNING	Processor running, is not idle and is executing instructions
2	HALTED	External halt signal asserted
3	STANDBY_WFE	Last instruction executed was WFE and standby mode has been entered
4	STANDBY_WFI	Last instruction executed was WFI and standby mode has been entered
5	IN_RESET	External reset signal asserted
6	DORMANT	Partial processor power down

Value	State label	Description
7	SHUTDOWN	Complete processor power down

**getValue()**

```
master behavior getValue() : uint64_t;
```

Obtain the number of instructions executed by the processor.

## 2.30 KeyboardStatus protocol

Defined in `$PVLIB_HOME/LISA/KeyboardStatusProtocol.lisa`.

**About KeyboardStatus protocol**

This protocol passes keyboard events to a component such as the PS2Keyboard component.

Events are only sent when the visualization window is in focus. Keyboard combinations that are filtered by the host OS such as Ctrl+Alt+Del are not detected by the visualization.

See `$PVLIB_HOME/include/components/KeyCode.h` for a list of `ATKeyCode` values.

`KeyboardStatus` provides the following behaviors:

**keyDown()**

```
slave behavior keyDown(ATKeyCode code) : void;
```

Sent when a key on the host keyboard is pressed.

**keyUp()**

```
slave behavior keyUp(ATKeyCode code) : void;
```

Sent when a key on the host keyboard is released.

## 2.31 LCD protocol

Defined in `$PVLIB_HOME/LISA/LCDPort.lisa`.

**About LCD protocol**

This Visualisation Library signaling protocol provides the interface between an LCD controller peripheral, for example the PL110, and a visualization component. This permits the LCD controller to render the framebuffer contents into a region of the visualization GUI.



LISA visualization components can provide any number of LCD ports. The implementations of these behaviors can delegate the calls to appropriate methods on the `VisRenderRegion` class.

`LCD` provides the following behaviors:

### **lock()**

```
slave behavior lock() : const VisRasterLayout*;
```

Lock the raster region, ready for rendering onto.

### **setPreferredLayout()**

```
slave behavior setPreferredLayout(unsigned int width, unsigned int height, unsigned  
int depth) : void;
```

Set the preferred pixel size and bitdepth of the LCD panel.

### **unlock()**

```
slave behavior unlock() : void;
```

Unlock the raster region, ready to update.

### **update()**

```
slave behavior update(int x, int y, unsigned int w, unsigned int h) : void;
```

Update part of the render region onto the screen.

## 2.32 LCDLayoutInfo protocol

Defined in `$PVLIB_HOME/examples/LISA/Common/LISA/LCDLayoutInfoProtocol.lisa`.

### About LCDLayoutInfo protocol

This protocol has the behavior `setLayoutInfo`.

`LCDLayoutInfo` provides the following behaviors:

### **setLayoutInfo()**

```
slave behavior setLayoutInfo(int x, int y, uint32_t w, uint32_t h) : void;
```

Sets the width and height of the touchscreen.

## 2.33 MMC\_Protocol protocol

Defined in `$PVLIB_HOME/LISA/MMC_Protocol.lisa`.

### About MMC\_Protocol

This protocol describes an abstract, untimed interface between an MMC controller and an MMC or SD card.

The protocol contains methods that must be implemented by the master (controller) or by the slave (card). This protocol is used by the reference PL180 MCI and MMC models. For further information on the protocol implementation, see the source file `$PVLIB_HOME/LISA/MMC_Protocol.lisa`.

Use of this protocol assumes knowledge of the MultiMediaCard specification, available from the [MultiMediaCard Association](#).

`MMC_Protocol` provides the following behaviors:

#### **Rx()**

```
master behavior Rx(const uint8_t *block, uint32_t len) : bool;
```

Read behaviours, from the card to the controller.

After the controller has issued a block or multiple block read command, the card calls the controller's `Rx()` method, with the first block. When the controller has consumed the block, that is, when it is able to accept another block, it should inform the slave with an `Rx_rdy()` call.

The slave might not provide a block immediately. It might wait until the controller is ready and the simulated transfer rate limits have been satisfied. This is important to avoid swamping the simulation with a large transfer at the expense of all other simulation activity.

The master might signal that it was not able to accept the given block, by returning false from `Rx()`. This is effectively a protocol error, and the card may retransmit the block later, or fail.

#### **Rx\_rdy()**

```
slave behavior Rx_rdy(void) : void;
```

#### **Tx()**

```
master behavior Tx(uint8_t *block, uint32_t len) : bool;
```

Write behaviours, from the controller to the card.

To minimize the number of times written data are copied, the following protocol is somewhat counter-intuitive. The basic premise is that an MMC controller usually contains a small data FIFO, which is filled either by the simulated CPU, or more frequently by DMA. The DMA typically occurs word by word. An efficient approach is therefore to construct the controller such that it can write directly into a buffer of stored card data.

When the write command is issued, the card calls the master with a pointer to the block that needs to be written. The master can then fill the block, calling `Tx_done()` when the block has been transferred. The card is again responsible for throttling to a simulated transfer rate, and will respond at some time in the future by providing another block to be written by calling the controller's `Tx()` function.

This approach has some drawbacks:

- Some timing and controller behavioral accuracy is sacrificed
- The controller might need to buffer data before a block is provided, if it cannot prevent data coming into its FIFO.

### **`Tx_done()`**

```
slave behavior Tx_done(void) : void;
```

### **`cmd()`**

```
slave behavior cmd(mmc_cmd_t cmd, uint32_t arg, void *resp) : mmc_resp_t;
```

The controller can send the slave a command, with an optional 32-bit argument.

The master must send in a void pointer to 128 bits of data.

The slave responds with a response type, and fills in up to 128 bits with data.

The master can check that the response type matches expectations, but this should not be necessary.

CRC is not implemented and start/stop bits are unnecessary at this level.



This behavior is not re-entrant by current design.

---

### **`cmd_name()`**

```
slave behavior cmd_name(mmc_cmd_t cmd) : const char*;
```

The slave implements this behavior to return a string for a given command. This is not strictly part of the MMC protocol.

## 2.34 MMU\_400\_BASE\_IDENTIFY protocol

Defined in \$PVLIB\_HOME/LISA/SMMU\_400\_BASE.lisa.

MMU\_400\_BASE\_IDENTIFY provides the following behaviors:

### identify()

```

slave behavior identify(
    const pv::TransactionAttributes* attributes_,
    bool is_read_,
    unsigned* stream_id_,
    unsigned* ssd_or_ssd_index_
) : void;

```

The way that the MMU-400 is configured to generate the `streamID` and `ssd_index` is complicated and must be done by implementing this function. This knowledge is specific to the SoC and to the devices generating the transactions and so it is not easily parameterisable.

## 2.35 MMU\_400\_Internals protocol

Defined in \$PVLIB\_HOME/LISA/SMMU\_400\_BASE.lisa.

### About MMU\_400\_Internals protocol

This protocol is for probing the internals of the MMU\_400. It has no correspondence in hardware. It is only intended for testing and informational purposes.

MMU\_400\_Internals provides the following behaviors:

### getMMU\_400()

```

slave behavior getMMU_400() : MMU_400::mmu_400_if*;

```

## 2.36 MMU\_500\_BASE\_IDENTIFY protocol

Defined in \$PVLIB\_HOME/LISA/SMMU\_500\_BASE.lisa.

MMU\_500\_BASE\_IDENTIFY provides the following behaviors:

### identify()

```

slave behavior identify(
    const pv::TransactionAttributes* attributes_,
    bool is_read_,
    unsigned tbu_number_,
    unsigned* stream_id_,
    unsigned* ssd_or_ssd_index_
) : void;

```

The way that the MMU-500 is configured to generate the StreamID and SSD\_Index or SSD is complicated and must be done by implementing this function. This knowledge is specific to the SoC and to the devices generating the transactions and so it is not easily parameterisable.

Note that the LACr0/r1 RTL encodes the TBU number into the bits [14:10] of the StreamId and the SSD\_Index, if being used. The bottom 10 bits are determined from the incoming transaction. In the LACr0/r1 RTL, each TBU can have fewer than 10 bits of `streamId/ssd_Index`, in which case they are zero-extended before being placed into bits[9:0].

The width of the TBU ID busses is invisible to the programmer and does not have an effect in the model except that this `identify()` function must obey the SoC's configuration.

For LACr0/r1, the caller automatically puts the TBU number in the `ids` itself. The callee is also allowed to do this, but the result is checked by an `assert()`, otherwise just leave these bits as zero.

For EAC, no such check is made and the platform must supply all 15 bits.

If you are supplying an SSD directly, and set the parameter `use_ssd_determination_table` to false so that `SMMU_IDR1.SSDTP == 0`, then the constants generated by `components/SMMU.h:ssd_secure()` and `ssd_non_secure()` should be used to return the SSD in `*ssd_or_ssd_index_`.

## 2.37 MMU\_500 Internals protocol

Defined in `$PVLIB_HOME/LISA/SMMU_500_BASE.lisa`.

### About MMU\_500 Internals protocol

This protocol is for probing the internals of the MMU\_500. It has no correspondence in hardware. It is only intended for testing and informational purposes.

`MMU_500_Internals` provides the following behaviors:

#### **getMMU\_500()**

```
slave behavior getMMU_500() : MMU_500::mmu_500_if*;
```

## 2.38 MouseStatus protocol

Defined in `$PVLIB_HOME/LISA/MouseStatusProtocol.lisa`.

### About MouseStatus protocol

This protocol passes mouse movement and button events to another component such as the PS2Mouse component.

Events are only sent when the visualization window is in focus.

`MouseStatus` provides the following behaviors:

### **mouseButton()**

```
slave behavior mouseButton(uint8_t button, bool down) : void;
```

This is sent when a button on the host mouse is pressed or released.

`button` indicates which button has been pressed or released and is typically 0, 1, or 2 but can be anything up to 7 depending on the OS and attached mouse.

`down` is true if a button is pressed and false if released.

### **mouseMove()**

```
slave behavior mouseMove(int dx, int dy) : void;
```

This is sent when the host mouse is moved. Mouse movement events are always relative.

## 2.39 PASSwitchControl protocol

Defined in `$PVLIB_HOME/LISA/PASSwitch.lisa`.

### About PASSwitchControl protocol

Allow transactions from the RME world (realm/pas/secure/non\_secure) to be routed separately.

Transactions for the RME PAS worlds are, by default, routed through the manager port `pvbus_m[PAS-value]`, where `PAS-value` is:

- |          |            |
|----------|------------|
| <b>0</b> | Secure     |
| <b>1</b> | Non-secure |
| <b>2</b> | Root       |
| <b>3</b> | Realm      |

The control port allows each PAS world to be one of the following:

- Routed to any of the four manager ports `pvbus_m[0]...pvbus_m[3]`
- Ignored
- Aborted

`PASSwitchControl` provides the following behaviors:

**routeAccessesForRmeWorlds()**

```

slave behavior routeAccessesForRmeWorlds(
    pv::PASSwitch_RouteOption route_secure,
    pv::PASSwitch_RouteOption route_non_secure,
    pv::PASSwitch_RouteOption route_root,
    pv::PASSwitch_RouteOption route_realm,
    pv::PASSwitch_RouteOption route_system_agent,
    pv::PASSwitch_RouteOption route_non_secure_protected) : void;

```

The arguments to the control port behavior `routeAccessesForRmeWorlds()` select how the chosen transactions are routed. They can have the following values:

**PORT\_IGNORE**

Transactions are ignored. Reads return 0.

**PORT\_ABORT**

Cause transactions to generate an abort.

**PORT\_0**

Route transactions to `pvbus_m[0]`.

**PORT\_1**

Route transactions to `pvbus_m[1]`.

**PORT\_2**

Route transactions to `pvbus_m[2]`.

**PORT\_3**

Route transactions to `pvbus_m[3]`.

**PORT\_4**

Route transactions to `pvbus_m[4]`.

**PORT\_5**

Route transactions to `pvbus_m[5]`.

Initial routing is configured using these PASSwitch parameters:

- `secure_port_index`
- `non_secure_port_index`
- `root_port_index`
- `realm_port_index`

Both default and explicit parameter values are overridden by runtime calls to `routeAccessesForRmeWorlds()` on the control port.

## 2.40 PCIDevice2ClientProtocol protocol

Defined in \$PVLIB\_HOME/LISA/PCIDevice2ClientProtocol.lisa.

### About PCIDevice2ClientProtocol protocol

This is a private protocol between PCIDevice and its wrapped client device.

PCIDevice2ClientProtocol provides the following behaviors:

#### **check\_if\_msix\_is\_enabled()**

```
optional master behavior check_if_msix_is_enabled() : bool;
```

A request from the client device to know whether MSI-X interrupt generation capability is enabled for the endpoint.

#### **generate\_MSI\_X()**

```
master behavior generate_MSI_X(
    unsigned          vector_index_,
) : int;
```

A request from the client device to map an MSI vector address to an address and data. Returns false if no MSI should be generated.

Return values:

- 1**  
Abort
- 0**  
Suppressed
- 1**  
OK

#### **get\_PRI\_client\_interface()**

```
slave behavior get_PRI_client_interface() : pcie_client_device_pri_if*;
```

If the device has PRI capability, it can adjust how the ATC makes PRI requests by implementing this interface.

#### **get\_transaction\_monitor\_control\_if()**

```
optional master behavior get_transaction_monitor_control_if(
    pcie::pcie_transaction_monitor_client_if* client_if_
) : pcie::pcie_transaction_monitor_control_if*;
```

A protocol which a client device can use to get the transaction monitor control interface implemented by the endpoint. The client device does this by requesting the interface using



`get_transaction_monitor_control_if()`. While requesting, the client device can pass a pointer to its own interface, which then provides an interface from the endpoint to the client.

### **identify()**

```
optional slave behavior identify(const pv::RemapRequest& req_, uint32_t*
    substreamid_) : void;
```

If the client can produce substreamids, it must use this behavior to fill `substreamid_`. If no substreamid is present on the request represented by `req_` then it should be assigned to `~0u`.

### **log\_error()**

```
optional master behavior log_error(
    pcie_service::ErrorMessage::ErrorCode_t error_code_,
    pcie_service::pcie_aer_error_type_t error_type_) : void;
```

A request to log a client device error with a specific error message.

## 2.41 PCIeATC\_get\_if protocol

Defined in `$PVLIB_HOME/LISA/PCIeATC.lisa`.

`PCIeATC_get_if` provides the following behaviors:

### **get\_if()**

```
slave behavior get_if() : pcie_atc_if*;
```

## 2.42 PChannel protocol

Defined in `$PVLIB_HOME/LISA/PChannelProtocol.lisa`.

### **About PChannel protocol**

Communicates power state changes between a power controller and a device.

You can use PChannels to replace `STANDBYWFI` and `STANDBYWFE` signaling.

For example, using `STANDBYWFI` OR `STANDBYWFE`:

- Core drives `STANDBYWFI` signal HIGH.
- Power controller performs logic x.

Equivalent behavior using PChannels:

- Core calls `pactive(OFF)`.
- Power controller calls `prequest(OFF)` to change the core to OFF.

- Power controller performs logic x.
- To wake up the core, the power controller calls `prequest (ON)`.

## Examples

- For a LISA+ example that uses PChannel, see `$PVLIB_HOME/examples/LISA/VP_PChannel/`.
- For a SystemC example that uses PChannel, see `$PVLIB_HOME/examples/SystemCExport/EVS_Components/EVS_PChannel/`.

`PChannel` provides the following behaviors:

### **`pactive()`**

```
master behavior pactive (uint32_t pstate) : void;
```

This master behavior is implemented by a power controller. A device calls this method to give a hint to the power controller that it can change to a particular power state. A power controller can then take appropriate action, typically communicating with the device by calling `device.prequest(new_power_state)`.

The power state is type `uint32_t` because it is the responsibility of the system using PChannels to enumerate the power states that it supports. For example, Armv8-A cores use the following enumeration for power states:

```
enum { OFF = 0,
      OFF_EMU,
      MEM_RET,
      MEM_RET_EMU,
      LOGIC_RET,
      FULL_RET,
      MEM_OFF,
      FUNC_RET,
      ON,
      WARM_RST,
      DBG_RECOV }
```

### **`prequest()`**

```
slave behavior prequest (uint32_t pstate) : sg::PChannel::presp_t;
```

This slave behavior is implemented by a device, for instance a core. A power controller typically calls this method and checks for the response from the device, which can either be `ACCEPT` or `DENY`.

The `sg::PChannel::presp_t` enumeration provides two values, `ACCEPT` and `DENY`. It is returned by the `prequest()` method, depending on the state requested and the current state of the core.

## 2.43 PL080\_DMAC\_DmaPortProtocol protocol

Defined in `$PVLIB_HOME/LISA/PL080_DMAC_DmaPortProtocol.lisa`.

### About PL080\_DMAC\_DmaPortProtocol protocol

The DmaPortProtocol is used to communicate handshake signals between the PL080\_DMAC controller and other peripherals in the system.

Depending on the PL080\_DMAC configuration, the `PL080_RES_CLR` signal might not be used.

`PL080_DMAC_DmaPortProtocol` provides the following behaviors:

#### **request()**

```
slave behavior request(uint32_t request) : void;
```

Passes requests from a peripheral to the DMA controller. The request is a bitfield with the low four bits defined. The request is level-sensitive and latched internally by the DMA controller. It is sampled and interpreted in a manner dependent on the target channel and configured flow control. It can have one of the following values:

**1**

`PL080_REQ_BURST`. Burst transfer request.

**2**

`PL080_REQ_SINGLE`. Single transfer request.

**4**

`PL080_REQ_LBURST`. Last burst request.

**8**

`PL080_REQ_LSINGLE`. Last single request.

#### **response()**

```
master behavior response(uint32_t response) : void;
```

Passes responses from the DMA controller to the peripherals. The response is a bitfield with the low two bits defined. It is transient rather than level-sensitive:

**1**

`PL080_RES_TC`. Terminal count response.

**2**

`PL080_RES_CLR`. Clear request response.

## 2.44 PL330\_DMAC\_DmaPortProtocol protocol

Defined in `$PVLIB_HOME/LISA/PL330_DMAC_DmaPortProtocol.lisa`.

### About PL330\_DMAC\_DmaPortProtocol protocol

The DmaPortProtocol is used to communicate handshake signals between the PL330\_DMAC controller and other peripherals in the system.

Depending on the PL330\_DMAC configuration, the PL330\_RES\_CLR signal may not be used.

PL330\_DMAC\_DmaPortProtocol provides the following behaviors:

#### **request()**

```
slave behavior request(uint32_t request) : void;
```

Requests from the external peripheral to the DMA controller. These are level-sensitive and are sampled by the DMA controller at specific points during the handshake. See the PL330\_DMAC.lisa implementation for more details.

#### **response()**

```
master behavior response(uint32_t response) : void;
```

Responses from the DMA controller to the external component. These are transient.

## 2.45 PMUEvent protocol

Defined in `$PVLIB_HOME/LISA/CCIRegisters.lisa`.

PMUEvent provides the following behaviors:

#### **fire()**

```
peer behavior fire() : void;
```

Trigger a PMU event.

## 2.46 PS2Data protocol

Defined in `$PVLIB_HOME/LISA/PS2DataProtocol.lisa`.

### About PS2Data protocol

This protocol is for communication between the Keyboard/Mouse Interface (KMI) and a PS/2-like device.

For efficiency, the interface is a parallel byte interface rather than a serial clock/data interface.

PS2Data provides the following behaviors:

### **getData()**

```
slave behavior getData () : uint8_t;
```

Used by the PS/2 device to get command data from the KMI.

### **putData()**

```
slave behavior putData (uint8_t data) : void;
```

Used by the PS/2 device to send device data to the KMI.

### **setClockData()**

```
master behavior setClockData (enum ps2clockdata_state) : void;
```

Used by the KMI to simulate forcing the state of the data/clock lines, to indicate whether it:

- Is able to receive data
- Wants to send a command
- Is inhibiting communication

## 2.47 PVBus protocol

Defined in \$PVLIB\_HOME/LISA/PVBusProtocol.lisa.

### About PVBus protocol

PVBus is used to provide bus connections for PV core models, or for any user-defined bus masters, to a tree of bus decoders and bus slave devices.

The bus protocol is designed to allow efficient calling through the bus decode tree, but it also implements back doors that allow bus masters to cache the decode results and access devices directly.

A bus slave component must instantiate a PVBusSlave subcomponent to provide an end point for the bus. The PVBusSlave component encapsulates all the complexity of handling the internal PVBus protocol. The PVBusSlave can be configured to handle all incoming transactions, see the example below, or as a bridge to the public PVDevice protocol.

Example of using PVBus for efficient access to memory-like storage:

```
component MemorySlave      // A component containing 64 MB of fast RAM
{
    slave port<PVBus> pvbus_s;
```

```

master port <PVBusSlaveControl> bus_slave_control;

composition
{
    bus_slave : PVBusSlave(size = 0x04000000);
}
connection
{
    self.pvbus_s => bus_slave.pvbus_s;
    self.bus_slave_control => bus_slave.control;
}
behavior init()
{
    bus_slave_control.setAccess(0, 0x04000000, pv::ACCESSTYPE_RW,
pv::ACCESSMODE_MEMORY);
    composition.init();
}
}

```



The following behaviors described as internal are implemented internally by PVBusMaster and PVBusSlave. Devices should not implement them.

PVBus provides the following behaviors:

#### **aceSnoopRequest()**

```
optional master behavior aceSnoopRequest( ACE::SnoopRequest* ) : void;
```

Internal behavior to support a coherency request from downstream.

#### **busMapChanged()**

```
optional master behavior busMapChanged(pv::bus_addr_t base, pv::bus_addr_t size) :
void;
```

Internal behavior used to handle cached bus decodings.

#### **debugACESnoopRequest()**

```
optional master behavior debugACESnoopRequest( ACE::SnoopRequest* ) : void;
```

Internal behavior to support a coherency request from downstream.

#### **debugRead()**

```
slave behavior debugRead(pv::ReadTransaction tx) : pv::Tx_Result;
```

Device access behavior for the PVBus protocol.

**Note**

Use of this behavior is deprecated. Use a PVBUSMaster and PVBUSSlave to send and receive PVBUS transactions.

### **debugWrite ()**

```
slave behavior debugWrite(pv::WriteTransaction tx) : pv::Tx_Result;
```

Device access behavior for the PVBUS protocol.

**Note**

Use of this behavior is deprecated. Use a PVBUSMaster and PVBUSSlave to send and receive PVBUS transactions.

### **discoverDownstreamChildDVMNodes ()**

```
optional slave behavior discoverDownstreamChildDVMNodes(DVM::DownstreamVisitor *) :  
void;
```

Internal behavior to support DVM message passing.

Allow a PVBUS master to probe a bus port for any slaves that can propagate DVM messages.

Bus routing fabric should forward the discovery request to all slaves.

### **discoverUpstreamParentDVMNodes ()**

```
optional master behavior discoverUpstreamParentDVMNodes(DVM::UpstreamVisitor *) :  
void;
```

Internal behavior to allow a PVBUS slave to probe a bus port for any masters that can respond to DVM messages.

### **doReadAccess ()**

```
optional slave behavior doReadAccess (pv::ReadRequest *) : pv::Tx_Result;
```

Internal behavior to support PVBUS re-entrant channels.

### **doWriteAccess ()**

```
optional slave behavior doWriteAccess (pv::WriteRequest *) : pv::Tx_Result;
```

Internal behavior to support PVBUS re-entrant channels.

**read()**

```
slave behavior read(pv::ReadTransaction tx) : pv::Tx_Result;
```

Device access behavior for the PVBUS protocol.



Use of this behavior is deprecated. Use a PVBUSMaster and PVBUSSlave to send and receive PVBUS transactions.

**write()**

```
slave behavior write(pv::WriteTransaction tx) : pv::Tx_Result;
```

Device access behavior for the PVBUS protocol.



Use of this behavior is deprecated. Use a PVBUSMaster and PVBUSSlave to send and receive PVBUS transactions.

## 2.48 PVBUS2PCI2PCIDeviceProtocol protocol

Defined in \$PVLIB\_HOME/LISA/PVBUS2PCI2PCIDeviceProtocol.lisa.

### About PVBUS2PCI2PCIDeviceProtocol protocol

This is the protocol between the PVBUS2PCI and PCIDevice components.

It is used to aggregate the many connections between the two.

PVBUS2PCI2PCIDeviceProtocol provides the following behaviors:

**get\_device\_assignment\_info()**

```
optional slave behavior get_device_assignment_info(sg::device_assignment_info_t&
da_info_) : void;
```

**get\_port\_info()**

```
slave behavior get_port_info(sg::port_info_t& out_) : void;
```

PCIDevice fills out\_ with its information.



**get\_port\_number()**

```
optional slave behavior get_port_number() : uint32_t;
```

**get\_selective\_reg\_block\_info()**

```
optional slave behavior get_selective_reg_block_info() :  
sg::ide::selective_stream_ide_reg_block_info_t;
```

A method to get the Selective Stream register block information. The information is used at the rootport to check the transactions attributes with respect to rootport DA conditions.

**get\_send\_error\_to\_rcec\_if()**

```
optional slave behavior get_send_error_to_rcec_if(  
std::vector<uint32_t>& rciep_device_function_table,  
uint32_t& bdf) : pcie::send_error_to_rcec_if*;
```

A method to check whether a downstream device is an RCEC device or not.

If so, get the pointer to pvbus2pci to RCEC if, which can be used to route error messages received in PVBUS2PCI/RC towards RCECs.

It also captures RCEC associated RCiEP's device-function information to check whether the device-function passed to RCEC points to a valid RCiEP.

If not implemented or returns nullptr then the device is not an RCEC.



**Note**

This is called during the reset phase, so the endPoint must cope if its own reset phase has not yet been called.

The first argument indicates the vector of device\_function\_info of RCEC associated RCiEPs. The second argument is the BDF of the downstream RCEC device. This is an optional behaviour.

**respond\_if\_address\_is\_captured()**

```
optional master behavior respond_if_address_is_captured(pv::bus_addr_t address) :  
bool;
```

A method implemented in bridge-type devices to query whether a given address will be routed downstream to it.

**set\_bus\_properties()**

```
optional slave behavior set_bus_properties(sg::pcie_bus_properties_t&) : void;
```

The properties of the bus can change dynamically and multiple calls to this behaviour should be expected.

## 2.49 PVBusBridgeControl protocol

Defined in `$PVLIB_HOME/LISA/PVBusBridgeControlProtocol.lisa`.

### About PVBusBridgeControl protocol

Allow a component to control its PVBusBridge subcomponent.

PVBusBridgeControl provides the following behaviors:

#### configure()

```
slave behavior configure(pv::slave_config_t*) : void;
```

Allow configuration of the transactions that are accepted by a PVBusBridge. By default the bridge accepts read and write transactions.

The `slave_config_t` class provides the following methods to extend the set of accepted transactions:

- `acceptACE_CleanShared_CleanInvalid_MakeInvalid()`
- `acceptACE_CleanUnique_MakeUnique()`
- `acceptEvict()`
- `acceptMemoryBarriers()`
- `acceptPrefetchOnly()`
- `acceptExclusiveTransactions()`

#### revokePrefetch()

```
slave behavior revokePrefetch(pv::bus_addr_t base, pv::bus_addr_t top,  
pv::DirectMemPtrData::AccessMask denied_access) : void;
```

Invalidates a DMI access range. `base` and `top` are included in the range.

## 2.50 PVBusCacheControl protocol

Defined in `$PVLIB_HOME/LISA/PVBusCache.lisa`.

### About PVBusCacheControl protocol

This protocol defines behaviors that are private, subject to change, and should not be used outside of the PVBusCache component.

PVBusCacheControl provides the following behaviors:

**createTransactionGenerator()**

```
slave behavior createTransactionGenerator(unsigned output_port) :  
    pv::TransactionGenerator*;
```

Get a transaction generator on the given output port.

**getLineContentsForRead()**

```
slave behavior getLineContentsForRead(unsigned line_index) : const char*;
```

Get temporary read access to the line data managed by `PvBusCache`. The line must have already been initialised by calling `getLineContentsForWrite`.

**getLineContentsForWrite()**

```
slave behavior getLineContentsForWrite(unsigned line_index) : char*;
```

Get temporary write access to the line data managed by `PvBusCache`. Allocates new storage for lines as needed.

**invalidateLineHit()**

```
slave behavior invalidateLineHit(unsigned hit_line_index,  
                                pv::CacheRevocation revoke_type) : void;
```

Revoke a line that has been marked as hitting.

**passThroughRead()**

```
slave behavior passThroughRead(unsigned output_port,  
                               pv::ReadTransaction tx) : pv::Tx_Result;
```

Pass through an unmodified read request. If a burst transaction spans more than one line, this only handles one line's worth of the burst.

**passThroughWrite()**

```
slave behavior passThroughWrite(unsigned output_port,  
                                pv::WriteTransaction tx) : pv::Tx_Result;
```

Pass through an unmodified write request. If a burst transaction spans more than one line, this only handles one line's worth of the burst.

**readFromLine()**

```
slave behavior readFromLine(pv::ReadTransaction tx,  
                            unsigned hit_line_index) : pv::Tx_Result;
```

Mark the current read transaction as hitting a cache line. All future transactions with the same attributes may be handled efficiently by `PVBusCache`, rather than being sent to the device `cacheRead()` or `cacheWrite()` handlers.

### **revokeRoutingDecisions()**

```
slave behavior revokeRoutingDecisions() : void;
```

Revoke all responses given by the `routeTransaction()` callback.

### **setTimingAnnotationConfig()**

```
slave behavior setTimingAnnotationConfig(pv::PVBusCacheTAConfig cfg) : void;
```

Set the timing annotation parameters.

### **writeToLine()**

```
slave behavior writeToLine(pv::WriteTransaction tx,
                           unsigned hit_line_index) : pv::Tx_Result;
```

Mark the current write transaction as hitting a cache line. All future transactions with the same attributes may be handled efficiently by `PVBusCache`, rather than being sent to the device `cacheRead()` or `cacheWrite()` handlers.

### **writeToLineAndPassThrough()**

```
slave behavior writeToLineAndPassThrough(pv::WriteTransaction tx,
                                         unsigned hit_line_index,
                                         unsigned output_port) : pv::Tx_Result;
```

Write the transaction data into a cache line, but also pass it through to a slave port. If a burst transaction spans more than one line, this only handles one line's worth of the burst.

## 2.51 PVBusCacheDevice protocol

Defined in `$PVLIB_HOME/LISA/PVBusCache.lisa`.

### About PVBusCacheDevice protocol

This protocol defines behaviors that are private, subject to change, and should not be used outside of the `PVBusCache` component.

`PVBusCacheDevice` provides the following behaviors:

### **cacheRead()**

```
slave behavior cacheRead(unsigned in_port,
                          pv::ReadTransaction tx) : pv::Tx_Result;
```

Handle a read request to the cache. For burst transactions, the cache can return after handling one line's worth of transaction data, and it is called back for the first beat on the next cache line.

### **cacheWrite()**

```
slave behavior cacheWrite(unsigned          in_port,
                          pv::WriteTransaction tx) : pv::Tx_Result;
```

Handle a write request to the cache.

### **routeTransaction()**

```
slave behavior routeTransaction(unsigned          in_port,
                                   pv::Transaction tx) : pv::CacheRoutingDecision;
```

Determine whether this transaction is cacheable. If not, decide which output port should forward the transaction.

## 2.52 PVBusMapperControl protocol

Defined in \$PVLIB\_HOME/LISA/PVBusMapperControlProtocol.lisa.

### About PVBusMapperControl protocol

Control protocol for use with PVBusMapper and PVBusModifier.

PVBusMapperControl provides the following behaviors:

#### **allBusMapChanging()**

```
master behavior allBusMapChanging() : void;
```

Something connected to the control port can generate an event to the upstream that indicates the bus map is changing and asks for all requests to be remapped again.

#### **getDVMNodesCanSendTo()**

```
master behavior getDVMNodesCanSendTo(
    std::vector<pv::DVMNodeRecord>& upstream_nodes_,
    std::vector<pv::DVMNodeRecord>& downstream_nodes_
) : bool;
```

To send DVM messages, you must have a description of where to send them.

You can ask PVBusMapper() to give you a vector of records containing all upstream and downstream nodes. This records set is only available after first reset. You pass in a vector that you want to be filled with the appropriate nodes.

It returns true if the lists are valid, even if `empty()`, otherwise it returns false and you should try again later. When the lists become valid, the expectation is that they remain valid and there is no need to call it again.

The `PVBusMapper` will always discover upstream and downstream DVM nodes. However, there may be a logical inconsistency if you use these records to send DVM messages if you are not handling DVM messages being send to you.

You may send a `DVMMessage` using the records returned.

It is expected that if you receive a DVM message and are forwarding it to other DVM nodes, you must take care not to forward it to the DVM node that gave you it. For this purpose, use the `getPortIndex()` and `getArcWithinPort()` methods and compare the results to the `port_index_` and `arc_within_port_` given to you by the `handle*DVMMessage()` calls.

### **getMyArcIdentifier()**

```
master behavior getMyArcIdentifier() : void*;
```

Return the arc identifier, `arc_within_port_`, that will be seen by a DVM node if we send a DVM message from this node.

### **handleDownstreamDVMMessageFromUpstream()**

```
optional slave behavior handleDownstreamDVMMessageFromUpstream(
    unsigned        upstream_port_index_,
    void*           arc_within_port_,
    DVM::Message*   message_
) : DVM::error_response_t;
```

If parameter `handling_of_dvm_messages_from_upstream` is set to `handle`, this behaviour is called when a DVM message from upstream is received.

You are given the `port_index_` that the DVM message came from and an opaque pointer to the upstream master within that connection. This pointer is the same one returned in the `DVMNodeRecord` obtained from `getDVMNodesCanSendTo()`.



Do not alter `message_` and forward it. You must first copy it and then forward the copy. The message might be in use by multiple components so altering it will also alter their version.



This is a message received from upstream, so it is a downstream DVM message.

---

## handleSnoopRequest()

```
optional slave behavior handleSnoopRequest(ACE::SnoopRequest* req_, bool debug_) :
    void;
```

Handle snoop requests. The `*SnoopRequest()` control port behaviors allow a `PVBusMapper` to act as an intermediary for snoop transactions on the bus.

If the `handling_of_upstream_snoop_requests` parameter is set to `handle`, this behavior is called when snoop transactions from any downstream port are received.



The snoop transaction is not automatically forwarded upstream but can be sent upstream using the `injectSnoopRequest()` behavior.

## handleUpstreamDVMMessageFromDownstream()

```
optional slave behavior handleUpstreamDVMMessageFromDownstream(
    unsigned        downstream_port_index_,
    void*           arc_within_port_,
    DVM::Message*   message_
) : DVM::error_response_t;
```

If parameter `handling_of_dvm_messages_from_downstream` is set to `handle`, this behaviour is called when a DVM message from downstream is received.

You are given the `port_index_` that the DVM message came from and an opaque pointer to the upstream master within that connection. This pointer is the same one returned in the `DVMNodeRecord` obtained from `getDVMNodesCanSendTo()`.



Do not alter `message_` and forward it. You must first copy it and then forward the copy. The message might be in use by multiple components so altering it will also alter their version.



This is a message received from downstream and so it is an upstream DVM message.

## injectSnoopRequest()

```
master behavior        injectSnoopRequest(ACE::SnoopRequest* req_, bool debug_) :
    void;
```

Issue a snoop transaction upstream.

**printDVMNodes()**

```
master behavior printDVMNodes(std::ostream&, const std::string& indent_) : void;
```

Print to the stream a text description of the nodes that it has currently found.

**remap()**

```
slave behavior remap(
    pv::RemapRequest& req_
) : unsigned;
```

Return the port that this transaction should be filtered to, based on the attributes and the address information held in the `RemapRequest` object. You may also indicate a remapping of the attributes and address in this call.

You may tag this decision with zero, one, or more objects of a type derived from `RemapDecisionGroup`. This allows you to revoke all decisions tagged with the same `RemapDecisionGroup` object. This object is allocated and owned by the component implementing the `remap()` function. See the `RemapDecisionGroup` class for more details.

The remapper must be consistent with respect to its decisions and so they must be statically determined.

The return value is a port number of `pvbuss_m`, or either of the special values:

**PVBUSMAPPER\_ABORT**

Abort all accesses

**PVBUSMAPPER\_IGNORE**

Treat all accesses as Read-As-Zero, Writes Ignored (**RAZ/WI**)

Any other value is considered an error.

**reset()**

```
master behavior reset() : void;
```

Signal a reset of the bus mapper bus interfaces. This is equivalent to an assert of the reset signal.

**sendAllBusMapChangingToUpstreamPort()**

```
master behavior sendAllBusMapChangingToUpstreamPort(
    unsigned upstream_port_index_
) : void;
```

Something connected to the control port can generate an event to an upstream port that indicates a bus map range is changing and asks for matching requests to be remapped again.



## 2.53 PVBUSOverTLMControl protocol

Defined in `$PVLIB_HOME/examples/SystemCExport/Bridges/PVBus2AMBAPVACE.lisa`.

### About PVBUSOverTLMControl protocol

This version of the PVBus to AMBA-PV bridge enables you to pass back all the coherency information from ACP.



This version makes use of a private and undocumented API that is not intended to be supported and will change in future releases.

PVBusOverTLMControl provides the following behaviors:

#### **routeAccesses ()**

```
slave behavior routeAccesses(BUS_RouteOption destination) : bool;
```

## 2.54 PVBUSRouterControl protocol

Defined in `$PVLIB_HOME/LISA/PVBusRouter.lisa`.

### About PVBUSRouterControl protocol

Allow the construction of arbitrary routing decisions.

PVBusRouterControl provides the following behaviors:

#### **filter ()**

```
slave behavior filter(  
    const pv::TransactionAttributes* attributes_,  
    pv::bus_addr_t                    page_base_,  
    bool                              is_read_  
) : unsigned;
```

Return the port that this transaction should be filtered to, based on the attributes and the `page_base_`, which is the address aligned to 4 KiB.

The filter must be consistent with respect to its filtering decisions and so they must be statically determined.

The return value is a port number of `pvbus_m`, or either of the special values:

#### **PVBUSROUTER\_ABORT**

Abort all accesses

**PVBUSROUTER\_IGNORE**

Treat all accesses as Read-As-Zero, Writes Ignored (**RAZ/WI**)

Any other value is considered an error.

## 2.55 PVBUSSlaveControl protocol

Defined in `$PVLIB_HOME/LISA/PVBUSSlaveControlProtocol.lisa`.

**About PVBUSSlaveControl**

Allow a component to configure its PVBUSSlave subcomponent.

This gives it control over mapping regions of device memory to be RAM, ROM, or device memory.

The PVBUSSlave automatically routes incoming bus accesses according to this configuration. Accesses to device memory, or writes to ROM memory, are routed to the device port, which the component should use to provide implementations of the `read()` and `write()` behaviors.

PVBUSSlaveControl provides the following behaviors:

**closeRegionIterHandle()**

```
slave behavior closeRegionIterHandle(uint32_t iter_handle) : void;
```

A caller may close an iterator opened by `getRegionIterHandle()` at any time using `closeRegionIterHandle()`. This deallocates the iterator and further uses of the handle are invalid.

**configure()**

```
slave behavior configure(pv::slave_config_t*) : void;
```

Allow configuration of the transactions that are accepted by a PVBUSSlave. By default the slave accepts read and write transactions.

The `slave_config_t` class provides the following methods to extend the set of accepted transactions:

- `acceptACE_CleanShared_CleanInvalid_MakeInvalid()`
- `acceptACE_CleanUnique_MakeUnique()`
- `acceptEvict()`
- `acceptMemoryBarriers()`
- `acceptPrefetchOnly()`
- `acceptExclusiveTransactions()`

**getNextRegionInfo()**

```
slave behavior getNextRegionInfo(uint32_t iter_handle,
                                pv::PVBUSSlaveRegionInfo *info) : bool;
```

After calling `getRegionIterHandle()`, the caller may repeatedly call `getNextRegionInfo()` with the provided `iter_handle`. If a region is found, the behavior returns true and the `info` struct is written to if the pointer is non-null. The region's data may be accessed using `getReadStorage()` or `getWriteStorage()`.

Regions may be returned in any order, and may be of any size or alignment, but no two regions overlap.

An implementation may decide not to report regions that have been allocated, but filled entirely with the default fill pattern, or regions allocated, but containing only the data they had at simulation start.

On reaching the last region, the iterator is automatically closed. If the handle is invalid or there are no further regions, the behavior returns false.

**getReadStorage()**

```
slave behavior getReadStorage(pv::bus_addr_t address,
                             pv::bus_addr_t *limit) : const uint8_t*;
```

Get read access to the underlying memory storage provided by the `PVBUSSlave`. The parameters are:

**address**

Byte address to request access to.

**limit**

Returns the address limit for the contiguous region.

The returned pointer can be used to directly access all memory locations from `address` to `limit-1`. The returned pointer is only guaranteed to remain valid until the next bus access or simulation cycle.

Modifying memory using `getWriteStorage()` does not inform any of the global exclusive monitors of the update.

**getRegionIterHandle()**

```
slave behavior getRegionIterHandle() : uint32_t;
```

An iterator-like API that allows a `PVBUSSlave` that provides storage to report all the regions of the address space that have backing store.

The iteration begins by calling `getRegionIterHandle()`. This allocates an iterator and if successful, returns a non-zero `iter_handle` to identify it.

**getWriteStorage()**

```
slave behavior getWriteStorage(pv::bus_addr_t address,
                              pv::bus_addr_t *limit) : uint8_t*;
```

Get write access to the underlying memory storage provided by the PVBUSSlave. The parameters are:

**address**

Byte address to request access to.

**limit**

Returns the address limit for the contiguous region.

The returned pointer can be used to directly access all memory locations from `address` to `limit-1`. The returned pointer is only guaranteed to remain valid until the next bus access or simulation cycle.

Modifying memory using `getWriteStorage()` does not inform any of the global exclusive monitors of the update.

**provideReadStorage()**

```
optional slave behavior provideReadStorage(pv::bus_addr_t device_base,
                                           pv::bus_addr_t device_limit,
                                           const uint8_t *storage) : void;
```

Allows a device to provide its own memory region to implement memory storage. For example, a device may want to ensure that the underlying memory is implemented in one contiguous region, or is allocated from a special region, for example video memory or memory-mapped memory.

The caller must allow the PVBUSSlave to take ownership of all accesses to this memory. In other words, the caller must call `getWriteStorage()` before modifying the contents of this memory region.

`device_base` and `device_limit` must be 4 KB-aligned.

Read latency for the range, which is used when Timing Annotation is enabled, is set from the PVBUSSlave `read_latency` parameter.

**provideReadStorageEx()**

```
slave behavior provideReadStorageEx(pv::bus_addr_t device_base,
                                    pv::bus_addr_t device_limit,
                                    const uint8_t *storage,
                                    double read_latency) : void;
```

This behavior is the same as `provideReadStorage()` but with an additional parameter to specify an average latency per byte, which is used when Timing Annotation is enabled.

**provideReadWriteStorage()**

```
optional slave behavior provideReadWriteStorage(
```

```
pv::bus_addr_t device_base,
pv::bus_addr_t device_limit,
uint8_t *storage) : void;
```

Allows a device to provide its own memory region to implement memory storage. For example, a device may want to ensure that the underlying memory is implemented in one contiguous region, or is allocated from a special region, for example video memory or memory-mapped memory.

The caller must allow the PVBUSSlave to take ownership of all accesses to this memory. In other words, the caller must call `getWriteStorage()` before modifying the contents of this memory region.

`device_base` and `device_limit` must be 4 KB-aligned.

Read and write latencies for the range, which are used when Timing Annotation is enabled, are set from the PVBUSSlave `read_latency` and `write_latency` parameters.

### **provideReadWriteStorageEx()**

```
slave behavior provideReadWriteStorageEx(
    pv::bus_addr_t device_base,
    pv::bus_addr_t device_limit,
    uint8_t *storage,
    double read_latency,
    double write_latency) : void;
```

This behavior is the same as `provideReadWriteStorage()` but with an additional parameter to specify an average latency per byte, which is used when Timing Annotation is enabled.

### **provideWriteStorage()**

```
optional slave behavior provideWriteStorage(pv::bus_addr_t device_base,
    pv::bus_addr_t device_limit,
    uint8_t *storage) : void;
```

Allows a device to provide its own memory region to implement memory storage. For example, a device may want to ensure that the underlying memory is implemented in one contiguous region, or is allocated from a special region, for example video memory or memory-mapped memory.

The caller must allow the PVBUSSlave to take ownership of all accesses to this memory. In other words, the caller must call `getWriteStorage()` before modifying the contents of this memory region.

`device_base` and `device_limit` must be 4 KB-aligned.

Write latency for the range, which is used when Timing Annotation is enabled, is set from the PVBUSSlave `write_latency` parameter.

### **provideWriteStorageEx()**

```
slave behavior provideWriteStorageEx(pv::bus_addr_t device_base,
    pv::bus_addr_t device_limit,
    uint8_t *storage,
```

```
double write_latency) : void;
```

This behavior is the same as `provideWriteStorage()` but with an additional parameter to specify an average latency per byte, which is used when Timing Annotation is enabled.

### **reset()**

```
slave behavior reset() : void;
```

Signal a reset of the bus slave interface.

This is equivalent to an assert of the reset signal.

### **setAccess()**

```
slave behavior setAccess(pv::bus_addr_t base,
                        pv::bus_addr_t top,
                        pv::accessType type,
                        pv::accessMode mode) : void;
```

Define how accesses are routed for a given range of device addresses. The parameters are:

#### **base**

Start address of the range to be configured, 4 KB-aligned.

#### **top**

End address, 4 KB-aligned.

#### **type**

Type of access to configure. Possible values:

- `ACCESSTYPE_READ`
- `ACCESSTYPE_WRITE`
- `ACCESSTYPE_RW`

#### **mode**

The new mode for accesses.

The following access modes control how to treat accesses of the selected type, within the chosen range:

#### **ACCESSMODE\_MEMORY**

Access data storage, which is managed by the PVBUSlave.

#### **ACCESSMODE\_DEVICE**

Route request to the device port on the slave.

#### **ACCESSMODE\_ABORT**

Generate an abort on the transaction.

#### **ACCESSMODE\_IGNORE**

Ignore the transaction. Reads return 0.

**setFillPattern()**

```
slave behavior setFillPattern (uint32_t fill1, uint32_t fill2) : void;
```

Set the default fill pattern for RAM or ROM regions. This should be called before any memory accesses occur, and allows memory to be prefilled with an alternating two-word pattern.

## 2.56 PVC2C protocol

Defined in \$PVLIB\_HOME/LISA/PVC2CProtocol.lisa.

**About PVC2C protocol**

PVC2C protocol models chip-to-chip connections. The data transfer in PVC2C is unidirectional, which is similar to CXS. Therefore, there is a dedicated interface for each transmitter and receiver.

pvc2c provides the following behaviors:

**discover\_pvc2c\_chips()**

```
slave behavior discover_pvc2c_chips(pvc2c::pvc2c_discovery_req_t*) : void;
```

**get\_haid()**

```
slave behavior get_haid() : pvc2c::haid_t;
```

**get\_pvc2c\_properties()**

```
slave behavior get_pvc2c_properties() : pvc2c::pvc2c_port_properties_t;
```

## 2.57 PVCacheDebugRam protocol

Defined in \$PVLIB\_HOME/LISA/PVCache\_DebugRamProtocol.lisa.

PVCacheDebugRam provides the following behaviors:

**getAttribute()**

```
slave behavior getAttribute(pv::PVCache_DebugRamPort::Attribute attribute, unsigned index) : uint64_t;
```

Get an attribute value from the cache.

This interface exposes certain numeric attributes of the cache. The parameters are:

**attribute**

Selects the attribute value to return. Must be one of the `A_*` enum constants specified in `PVCache_DebugRamPort.h`.

**index**

Currently unused and must be set to 0. It is intended to expose arrays of attributes, for example per-set/way attributes.

This behavior returns a numeric attribute value. It returns 0 for unknown or unsupported attributes or for index out of range.

**getConfig()**

```
slave behavior getConfig(pv::PVCache_DebugConfig& config) : void;
```

Obtain cache configuration, mainly geometry for now.

**getDeferredActions()**

```
slave behavior getDeferredActions() : sg_deferred_actions::deferred_actions_t*;
```

Get a handle to the `deferred_actions_t` object used by the implementation of the cache system.

The cache system has the ability to defer internal events until a re-entrant safe point by pushing them onto a `deferred_actions_t` object.

This method gets a handle to the object for validation purposes.

**peekLine()**

```
slave behavior peekLine(const pv::PVCache_DebugFilter& filter,  
pv::PVCache_DebugLine& buffer) : bool;
```

Peek the cache line location, tag, and content. The parameters are:

**filter**

Reference to a cache lookup filter. Filters can be constructed for lookup by address, index, set/way, and so on.

**buffer**

Reference to a buffer for the returned cache line location, tag, and content. If `buffer` is constructed with a zero length cache line, the cache line content is not returned.

If cache lookup is by address and fails to hit, the `location` member of `buffer` is updated with the first way in the cache where the cache line could have been.

This behavior returns the cache lookup/hit status. If false, the tag and the content in the buffer are not updated.



For lookup by index or set/way, if the selected line is within the boundary of the cache, the routine returns true and the `tag.valid` data member in `buffer` indicates whether the cache line, tag, and content members contain valid data.

For lookup by address, the return status indicates a hit or miss in the cache. A miss does not cause any further transactions downstream or allocation into the cache.

### **pokeLine()**

```
slave behavior pokeLine(const pv::PVCache_DebugFilter& filter, const
pv::PVCache_DebugLine& buffer) : bool;
```

Poke the cache line tag and content data.

The cache may not be able to accept all cache line tag modifications and may ignore some or all such modifications. But overwriting cache line content is always supported by the cache.

The parameters are:

#### **filter**

Reference to a cache lookup filter. Filters can be constructed for lookup by address, index, set/way and so on.

#### **buffer**

Reference to a buffer with cache line tag and content data to set. The cache line location member in `buffer` is ignored. If `buffer` is constructed with a zero length cache line, the cache line content is not set.

This behavior returns the cache lookup/hit status. For lookup by index or set/way, if the selected line is within the boundary of the cache, the routine returns true and the cache line state data inside the cache is modified.

For lookup by address, the return status indicates a hit or miss in the cache. A miss does not cause any further transactions downstream, allocation into the cache, or modification of cache line state.

## 2.58 PVCacheMaintenance protocol

Defined in `$PVLIB_HOME/LISA/PVCacheMaintenance.lisa`.

`PVCacheMaintenance` provides the following behaviors:

### **cacheSizeOverride()**

```
slave behavior cacheSizeOverride( unsigned cache_line_size_in_bytes_,
                                unsigned number_of_sets_,
                                unsigned number_of_ways_ ) : void;
```

Private internal functionality of the cache implementation. Do not use.

Override the cache size. All the contents of the cache will be lost at this point. If any lines have been allocated then ACE state may be corrupted across the system. The caller is responsible for ensuring that no transactions are in flight when calling this.

### **clean\_all()**

```
slave behavior clean_all() : void;
```

Clean the entire cache, flushing all dirty lines.

### **clean\_and\_invalidate\_all()**

```
optional slave behavior clean_and_invalidate_all() : void;
```

Clean and invalidate the entire cache, evicting all lines without cleaning.

### **clean\_and\_invalidate\_by\_addr()**

```
optional slave behavior clean_and_invalidate_by_addr(  
    pv::bus_addr_t addr, bool is_non_secure) : void;
```

Clean and invalidate by PA. This should be used in preference to `clean_by_addr()` followed by `invalidate_by_addr()` as a write could occur between the two and then the `invalidate_by_addr()` would invalidate dirty data.

### **clean\_and\_invalidate\_by\_addr\_by\_pas()**

```
optional slave behavior clean_and_invalidate_by_addr_by_pas(  
    pv::bus_addr_t addr, pv::PASpace_t pas) : void;
```

### **clean\_and\_invalidate\_by\_set\_way()**

```
optional slave behavior clean_and_invalidate_by_set_way(  
    uint32_t set, uint32_t way, bool is_non_secure) : void;
```

Clean and invalidate by set/way. Secure evicts any, non-secure only evicts non-secure entries.

### **clean\_and\_invalidate\_by\_set\_way\_by\_pas()**

```
optional slave behavior clean_and_invalidate_by_set_way_by_pas(  
    uint32_t set, uint32_t way, pv::PASpace_t pas) : void;
```

### **clean\_by\_addr()**

```
slave behavior clean_by_addr(pv::bus_addr_t addr, bool is_non_secure) : void;
```

Clean by PA, evicting the lines that match.

### **clean\_by\_addr\_by\_pas()**

```
optional slave behavior clean_by_addr_by_pas(  
    pv::bus_addr_t addr, pv::PASpace_t pas) : void;
```

```
pv::bus_addr_t addr, pv::PASpace_t pas) : void;
```

### **clean\_by\_set\_way()**

```
slave behavior clean_by_set_way(uint32_t set, uint32_t way, bool is_non_secure) :
void;
```

Clean by set/way. Secure evicts any, non-secure only evicts non-secure entries.

### **clean\_by\_set\_way\_by\_pas()**

```
optional slave behavior clean_by_set_way_by_pas(
uint32_t set, uint32_t way, pv::PASpace_t pas) : void;
```

### **enableLocalDVMMessageProcessing()**

```
slave behavior enableLocalDVMMessageProcessing( bool on_ ) : void;
```

Enable or disable whether the current cache handles DVM messages locally.

When the cache was created, there may have been the option of telling it to startup to ignore\_local\_dvm\_messages or not. This now makes that a dynamic behaviour.

### **enableUpstreamAcceptsDVM()**

```
slave behavior enableUpstreamAcceptsDVM( unsigned upstream_port_, bool on_ ) :
void;
```

Enable or disable which of the upstream ports are currently accepting DVM snoop requests.

This is used to override the local cache behavior.

### **enableUpstreamAcceptsSnoopRequests()**

```
slave behavior enableUpstreamAcceptsSnoopRequests( unsigned upstream_port_, bool
on_ ) : void;
```

Enable or disable which of the upstream ports are currently accepting ACE snoop requests.

This is used to override the local cache behavior.

### **find\_in\_cache()**

```
slave behavior find_in_cache(
const pv::MemoryAttributes &memory_attributes_,
pv::bus_addr_t address_
) : bool;
```

Test whether this layer of cache contains a given line.



The security world is encoded in the `memory_attributes_` parameter. The exclusive, cache maintenance, and debug flags are ignored.

### **getCacheStateModelled()**

```
slave behavior getCacheStateModelled() : bool;
```

Get the current value of “cache state modelled”.

### **getEnabled()**

```
slave behavior getEnabled(bool is_non_secure) : bool;
```

Get the enabled state.

### **getLockDown()**

```
slave behavior getLockDown() : uint32_t;
```

Get a bit array controlling which cache ways are locked down.

### **invalidate\_all()**

```
slave behavior invalidate_all() : void;
```

Invalidate the entire cache, evicting all lines without cleaning.

### **invalidate\_by\_addr()**

```
slave behavior invalidate_by_addr(pv::bus_addr_t addr, bool is_non_secure) : void;
```

Invalidate by PA, evicting the lines that match.

### **invalidate\_by\_addr\_by\_pas()**

```
optional slave behavior invalidate_by_addr_by_pas(
    pv::bus_addr_t addr, pv::PASpace_t pas) : void;
```

### **invalidate\_by\_set\_way()**

```
slave behavior invalidate_by_set_way(uint32_t set, uint32_t way, bool
    is_non_secure) : void;
```

Invalidate by set/way. Secure evicts any, non-secure only evicts non-secure entries.

### **invalidate\_by\_set\_way\_by\_pas()**

```
optional slave behavior invalidate_by_set_way_by_pas(
```

```
uint32_t set, uint32_t way, pv::PASpace_t pas) : void;
```

## preload()

```
slave behavior preload(
    const pv::MemoryAttributes &memory_attributes_,
    pv::bus_addr_t               address_,
    bool                         make_unique_,
    sg::ticks_t&                 local_time_
) : pv::Tx_Result;
```

Preload a line into this layer of the cache.

If you ask for it to be unique, it performs all the cache coherency operations to make it unique to that cache, assuming it is shared. This means that a write to that cache does not have to perform extra coherency operations, assuming it is still unique in the cache at that point. This is intended as a primitive to model preload for read and preload for write.



Note

The security world is encoded in the `memory_attributes_` parameter. The exclusive, cache maintenance, and debug flags are ignored.

## setBitmapOfDownstreamPortsThatIsDomainBoundaryForReallyNonShared()

```
optional slave behavior
setBitmapOfDownstreamPortsThatIsDomainBoundaryForReallyNonShared(uint64_t) : void;
```

Override the

`bitmap_of_downstream_ports_that_is_the_domain_boundary_for_really_non_shared` parameter.

This bitmap is used to indicate whether an nsh request treated as sh should be recovered back to nsh when it goes out to the downstream.

## setBitmapOfUpstreamPortsThatTreatNonSharedAsShared()

```
optional slave behavior setBitmapOfUpstreamPortsThatTreatNonSharedAsShared(uint64_t
    bitmap_of_upstream_ports_that_treat_nsh_as_sh) : void;
```

Override the `bitmap_of_upstream_ports_that_treat_non_shared_as_shared` parameter.

This bitmap is used to indicate that a non-shared request is treated as shared within the cluster or cache.

## setCacheStateModelled()

```
slave behavior setCacheStateModelled(bool modelled) : void;
```

Set the “cache state modelled” state.

**setEnabled()**

```
slave behavior setEnabled(bool enabled, bool is_non_secure) : void;
```

Set the enabled state.

**setIsInner()**

```
slave behavior setIsInner(bool is_inner) : void;
```

Set the domain for the cache as inner or outer.

**setLockDown()**

```
slave behavior setLockDown(uint32_t lock) : void;
```

Set a bit array controlling which cache ways are locked down.

**setNoDistinctionBetweenIshAndOsh()**

```
slave behavior setNoDistinctionBetweenIshAndOsh(bool  
no_distinction_between_ish_and_osh_) : void;
```

Reconfigure whether the cache treats the distinction between Inner-Shareability and Outer-Shareability as meaningful when matching attributes.



Note

The Outer/Inner Shareability distinction is preserved on the bus.



Note

The caller is responsible for ensuring that no transactions are in flight when calling this behaviour.

---

## 2.59 PVDevice protocol

Defined in \$PVLIB\_HOME/LISA/PVDeviceProtocol.lisa.

### About PVDevice protocol

Simple bus protocol that allows a LISA component to handle bus read/write transactions using a PVBusSlave subcomponent.

Examples of usage:

```
component SimpleSlave
{
  composition
  {
    bus_slave : PVBusSlave(size = 0x1000);
  }
  connection
  {
    self.pvbus_s => bus_slave.pvbus_s;
    bus_slave.device => self.device;
  }

  slave port <PVBus> pvbus_s;

  internal slave port<PVDevice> device
  {
    behavior read(pv::ReadTransaction tx) : pv::Tx_Result
    {
      switch(tx.getAddress() & ~3)
      {
        case 0: return tx.setReturnData32(0x12345678);
        default: return tx.generateAbort();
      }
    }
    behavior write(pv::WriteTransaction tx) : pv::Tx_Result
    {
      uint32_t data = tx.getData32();
      tx.writeComplete();
    }
    behavior debugRead(pv::ReadTransaction tx) : pv::Tx_Result
    {
      return device.read(tx);
    }
    behavior debugWrite(pv::WriteTransaction tx) : pv::Tx_Result
    {
      return device.write(tx);
    }
  }
}
```

PVDevice provides the following behaviors:

### **debugRead()**

```
slave behavior debugRead(pv::ReadTransaction tx) : pv::Tx_Result;
```

Enable the device to handle a debug read transaction.

### **debugWrite()**

```
slave behavior debugWrite(pv::WriteTransaction tx) : pv::Tx_Result;
```

Enable the device to handle a debug write transaction.

### **read()**

```
slave behavior read(pv::ReadTransaction tx) : pv::Tx_Result;
```

Enable the device to handle a bus read transaction.

### **revokePrefetch()**

```
master behavior revokePrefetch(pv::RevokeTransaction* tx,
    pv::range_t<pv::bus_addr_t> range) : void;
```

Allow the slave to revoke any prefetch information given to the master.

This revokes both read and write prefetches for the range given. The revoke transactions can be obtained using `tx.getPayload()->getRevokeTransaction()`.

See the equivalent behaviour in `PVBus.h` for more information.

### **write()**

```
slave behavior write(pv::WriteTransaction tx) : pv::Tx_Result;
```

Enable the device to handle a bus write transaction.

## 2.60 PVTransactionMaster protocol

Defined in `$PVLIB_HOME/LISA/PVTransactionMasterProtocol.lisa`.

### About PVTransactionMaster protocol

This protocol exists to allow bus masters to instantiate `TransactionGenerator` objects on the control port of a `PVBusMaster` subcomponent.

Any number of `TransactionGenerator` objects can be created from a single `PVBusMaster`.

They should be allocated at startup, because allocating a new `TransactionGenerator` for each transaction is expensive.

It is most efficient to have one `TransactionGenerator` for each data stream that is being accessed. For example, to get maximum efficiency from the `PVBus` system, a DMA memory transfer should use one generator for the reads and one for the writes.

The `TransactionGenerator` class is defined in `$PVLIB_HOME/include/pv/PVBusMaster.h`.

`PVTransactionMaster` provides the following behaviors:

### **createRandomContextTransactionGenerator()**

```
slave behavior createRandomContextTransactionGenerator() :
    pv::RandomContextTransactionGenerator*;
```

Return a new instance of a `RandomContextTransactionGenerator` object.



**createStreamingTransactionGenerator()**

```
slave behavior createStreamingTransactionGenerator() :
    pv::StreamingTransactionGenerator*;
```

Return a new instance of a `StreamingTransactionGenerator` object.

**createTransactionGenerator()**

```
slave behavior createTransactionGenerator() : pv::TransactionGenerator*;
```

Return a new instance of a `TransactionGenerator` object.

**reset()**

```
slave behavior reset() : void;
```

Signal a reset of the bus master interface. This is equivalent to a deassert of the reset signal.

## 2.61 PVWriteBuffer\_BarrierPort protocol

Defined in `$PVLIB_HOME/LISA/PVWriteBuffer.lisa`.

`PVWriteBuffer_BarrierPort` provides the following behaviors:

**CleanByAddr()**

```
slave behavior CleanByAddr(bus_addr_t addr, bool ns) : void;
```

**CleanByAddrNSNSE()**

```
slave behavior CleanByAddrNSNSE(bus_addr_t addr, bool ns, bool nse) : void;
```

**CleanByAddrPAs()**

```
slave behavior CleanByAddrPAs(bus_addr_t addr, pv::PASpace_t pas) : void;
```

**notify()**

```
slave behavior notify(PVWriteBufferComponentBarrier_t type) : void;
```

## 2.62 PVWriteBuffer\_SErrorPort protocol

Defined in `$PVLIB_HOME/LISA/PVWriteBuffer.lisa`.

`PVWriteBuffer_SErrorPort` provides the following behaviors:

**notify()**

```
slave behavior notify(const Tx_Result& result,
                     bus_addr_t address,
                     const Payload& payload,
                     const uint8_t* faultp,
                     PAspace_t pas) : void;
```

## 2.63 PVWriteBuffer\_VmidBarrierPort protocol

Defined in \$PVLIB\_HOME/LISA/PVWriteBuffer.lisa.

PVWriteBuffer\_VmidBarrierPort provides the following behaviors:

**notify()**

```
slave behavior notify(PVWriteBufferComponentBarrier_t type, unsigned vmid) : void;
```

## 2.64 SC\_ClockRateControl protocol

Defined in \$PVLIB\_HOME/examples/SystemCEExport/Common/Protocols/LISA/  
SC\_ClockRateControlProtocol.lisa.

**About SC\_ClockRateControl protocol**

Allow systems to dynamically modify the multiply/divide ratio of a ClockDivider component.

If a ClockDivider's ratio is changed, the frequency of its `clk_out` signal is immediately recalculated, along with any clocks derived from that signal.

Any active ClockTimers will automatically compute the number of ticks elapsed so far at the old clock rate, and continue counting down at the new rate. This may introduce a slight rounding error of a fraction of a tick.

SC\_ClockRateControl provides the following behaviors:

**set64\_m()**

```
master behavior set64_m(uint64_t mul, uint64_t div) : void;
```

Set clock rate. New clock rate =  $\text{mul} / \text{div}$ .

**set64\_s()**

```
slave behavior set64_s(uint64_t mul, uint64_t div) : void;
```

Set clock rate. New clock rate =  $\text{mul} / \text{div}$ .

**set\_m()**

```
master behavior set_m(uint32_t mul, uint32_t div) : void;
```

Set clock rate. New clock rate = mul / div.

**set\_s()**

```
slave behavior set_s(uint32_t mul, uint32_t div) : void;
```

Set clock rate. New clock rate = mul / div.

## 2.65 SC\_ClockSignal protocol

Defined in `$PVLIB_HOME/examples/SystemCExport/Common/Protocols/LISA/SC_ClockSignalProtocol.lisa`.

### About SC\_ClockSignal protocol

SystemC export equivalent of the LISA+ [ClockSignal protocol](#).

A ClockSignal port represents a timebase of a given frequency. This is an opaque port type. It contains no user-accessible behavior.

ClockSignal output ports are provided on the following library components:

#### MasterClock

Produces a clock signal at a base clock rate, which can nominally be considered to be 1Hz.

#### ClockDivider

Can be used to take an input ClockSignal from a MasterClock or from another ClockDivider and generate an output that is related to the input signal by a given ratio.

ClockSignals can be used as input to CpuComponents, to define the core clock rate. They can also be used to drive the clock port of a ClockTimer component, which can be used to generate events in the scheduler.



Note

A ClockSignal does not actually define a fixed square-wave signal. It merely defines a frequency that can be used by counter timers.

### Example system using ClockSignal

```
composition {
  masterclock : MasterClock;
  div_24MHz : ClockDivider(div = 1, mul = 24000000);
  timer : ClockTimer;
}
master port<TimerControl> timer_control;
slave port<TimerCallback> timer_callback {
```

```

    behavior signal() : uint32_t {
        // handle timed event here
        // ...
        // reschedule in 10 ticks of input clock
    }
}
behavior start_timer() {
    // start timer counting 10 ticks
}
connection {
    masterclock.clk_out  => div_24MHz.clk_in;
    div_24MHz.clk_out    => timer.clk_in;
    self.timer_control  => timer.timer_control;
    timer.timer_callback => self.timer_callback;
}

```

sc\_ClockSignal provides the following behaviors:

### **current\_ticks\_m()**

```
master behavior current_ticks_m() : uint64_t;
```

Private internal method used between Scheduler components.

### **current\_ticks\_s()**

```
slave behavior current_ticks_s() : uint64_t;
```

Private internal method used between Scheduler components.

### **get\_clock\_m()**

```
master behavior get_clock_m() : sg::FrequencySource*;
```

Private internal method used between Scheduler components.

### **get\_clock\_s()**

```
slave behavior get_clock_s() : sg::FrequencySource*;
```

Private internal method used between Scheduler components.

### **rate\_in\_hz\_m()**

```
master behavior rate_in_hz_m() : double;
```

Private internal method used between Scheduler components.

### **rate\_in\_hz\_s()**

```
slave behavior rate_in_hz_s() : double;
```

Private internal method used between Scheduler components.

**set\_clock\_m()**

```
master behavior set_clock_m(sg::FrequencySource* _sg_frequencysource_0) : void;
```

Private internal method used between Scheduler components.

**set\_clock\_s()**

```
slave behavior set_clock_s(sg::FrequencySource* _sg_frequencysource_0) : void;
```

Private internal method used between Scheduler components.

## 2.66 SC\_VirtualEthernet protocol

Defined in \$PVLIB\_HOME/examples/SystemCExport/Common/Protocols/LISA/SC\_VirtualEthernetProtocol.lisa.

### About SC\_VirtualEthernet protocol

SystemC equivalent of [VirtualEthernet protocol](#).

SC\_VirtualEthernet provides the following behaviors:

**send\_to\_master\_m()**

```
master behavior send_to_master_m(EthernetFrame* frame) : void;
```

**send\_to\_slave\_s()**

```
slave behavior send_to_slave_s(EthernetFrame* frame) : void;
```

## 2.67 SMMUv3AEMIdentifyProtocol protocol

Defined in \$PVLIB\_HOME/LISA/SMMUv3AEMIdentifyProtocol.lisa.

### About SMMUv3AEMIdentifyProtocol protocol

Architecturally, a transaction comes into the SMMU model with the following side band signals:

- Security State Determination (SSD):
  - 0 Transaction belongs to a device controlled by the secure world
  - 1 Transaction belongs to a device controlled by the non-secure world
  - 2 Transaction belongs to a device controlled by the root world

**3**

Transaction belongs to a device controlled by the realm world

- StreamID
- SubStreamID and SubStreamID valid

How these are transported in the system is SoC-dependent.

The SMMU model requires that the SoC provides a way of determining this information by providing the `identify()` behaviour.

SMMUV3AEMIdentifyProtocol provides the following behaviors:

**identify()**

```
slave behavior identify(
    unsigned                tbu_number_,
    const pv::TransactionAttributes* attributes_,
    bool*                   out_ssd_ns_,
    unsigned*               out_streamid_,
    unsigned*               out_substreamid_ // ~0u if no substreamid
) : void;
```

**identify\_2()**

```
optional slave behavior identify_2(
    unsigned                tbu_number_,
    const pv::TransactionAttributes* attributes_,
    unsigned*               out_ssd_, // 0 -- s, 1
    -- ns, 2 -- rt, 3 -- rl
    uint64_t*               out_streamid_, // ~0ull
    if NoStreamID
    unsigned*               out_substreamid_, // ~0u if
    no substreamid
    // Added between 11.16 and 11.17
    SMMUV3AEM::smmuv3aem_identify_protocol_extra_t* out_extra_ // For NoStreamID
    only
) : void;
```

## 2.68 SchedulerInterfaceControl protocol

Defined in `$PVLIB_HOME/LISA/SchedulerInterfaceControlProtocol.lisa`.

### About SchedulerInterfaceControl protocol

This protocol is used to access the Fast Models scheduler.

SchedulerInterfaceControl provides the following behaviors:

**waitTicks()**

```
slave behavior waitTicks(uint64_t ticks) : void;
```

Let the time of the calling thread advance by `ticks`, relative to `clk_in`.

## 2.69 SchedulerThreadControl protocol

Defined in `$PVLIB_HOME/LISA/SchedulerThreadControlProtocol.lisa`.

### About SchedulerThreadControl protocol

This protocol is used to control the behavior of the SchedulerThread component and also to run the actual thread code.

`SchedulerThreadControl` provides the following behaviors:

#### **setupThread()**

```
slave behavior setupThread(unsigned index, void *args, const
sg::SchedulerThreadParameters *parameters) : void;
```

Set up a new thread.

This function must only be called if more than one thread should be handled through this `SchedulerThread` instance. A default thread is always started with `index=0` and `args=0`.

Calling this function may or may not yield to other threads, not necessarily the newly-created thread.

Specifying `parameters = 0` has the same semantics as specifying a default constructed `SchedulerThreadParameters()`. The instance pointed to by `parameters` is not used after `setupThread()` returns.

#### **threadProc()**

```
master behavior threadProc(unsigned index, void *args) : void;
```

Actual thread function. The `index` and `args` parameters are set to 0, 0 for the default thread and are specified in the `setupThread(index args)` call for all additional threads.

#### **waitTicks()**

```
slave behavior waitTicks(uint64_t ticks) : void;
```

Let the time of this thread advance by `ticks`, relative to `clk_in`.

This is the same as `SchedulerInterfaceControl.waitTicks()`.

## 2.70 SchedulerThreadEventControl protocol

Defined in `$PVLIB_HOME/LISA/SchedulerThreadEventControlProtocol.lisa`.

### About SchedulerThreadEventControl protocol

This protocol is used to control the behavior of the `ThreadSignal` component.

`SchedulerThreadEventControl` provides the following behaviors:

#### `notify()`

```
slave behavior notify() : void;
```

Unblock any Fast Models threads waiting on this event.

Ignored if no threads are waiting. The event is not buffered until another thread tries to wait.

#### `wait()`

```
slave behavior wait() : void;
```

Block the current Fast Models thread, for example the `schedulerThread` instance, until anything calls `notify()`.

## 2.71 SerialData protocol

Defined in `$PVLIB_HOME/LISA/SerialData.lisa`.

### About SerialData protocol

This protocol is implemented as a parallel interface for efficiency. All communication is driven by the master port.

`serialData` provides the following behaviors:

#### `dataReceive()`

```
peer behavior dataReceive() : uint16_t;
```

Used by the master to receive data from the slave.

Table 2-3: Bits for `dataReceive()`

Bits	Function
15:13	Reserved
12	Set when no data available for reading
11	Reserved
10	Break error



Bits	Function
9:8	Reserved
7:0	Receive data

**dataTransmit()**

```
peer behavior dataTransmit(uint16_t data) : void;
```

Used by the master to send data to the slave.

**Table 2-4: Bits for dataTransmit()**

Bits	Function
15:8	Reserved
7:0	Transmit data

**signalsGet()**

```
peer behavior signalsGet() : uint8_t;
```

Used by the master to get the current signal status.

**Table 2-5: Bits for signalsGet()**

Bits	Function
7:4	Reserved
3	DCD
2	DSR
1	CTS
0	RI

**signalsSet()**

```
peer behavior signalsSet(uint8_t signal) : void;
```

Used by the master to set the current signal status.

**Table 2-6: Bits for signalsSet()**

Bits	Function
7	Out1
6	Out2
5	RTS
4	DTR
3:0	Reserved

## 2.72 Signal protocol

Defined in `$PVLIB_HOME/LISA/SignalProtocol.lisa`.

### About Signal protocol

The Signal protocol provides a single method that allows a master to set or clear a signal. This can be used for any level-sensitive signalling

The `sg::Signal::State` enumeration provides two values:

```
sg::Signal::Set  
sg::Signal::Clear
```

`signal` provides the following behaviors:

#### **setValue()**

```
peer behavior setValue(sg::Signal::State) : void;
```

Set signal value. Allowed values:

- `sg::Signal::Set`
- `sg::Signal::Clear`

## 2.73 StateSignal protocol

Defined in `$PVLIB_HOME/LISA/SignalProtocol.lisa`.

### About StateSignal protocol

The StateSignal protocol provides one method that allows a master to set or clear a signal and another allowing the master to retrieve the current state from a slave. This can be used for any level-sensitive signalling

The `sg::Signal::State` enumeration provides two values:

```
sg::Signal::Set  
sg::Signal::Clear
```

`stateSignal` provides the following behaviors:

#### **getValue()**

```
peer behavior getValue() : sg::Signal::State;
```

Returns the state of the signal.

**setValue()**

```
peer behavior setValue(sg::Signal::State) : void;
```

Set the signal value. Allowed values:

- `sg::Signal::Set`
- `sg::Signal::Clear`

## 2.74 SystemCCoprocBusProtocol protocol

Defined in `$PVLIB_HOME/examples/SystemCExport/Common/Protocols/LISA/SystemCCoprocBusProtocol.lisa`.

`SystemCCoprocBusProtocol` provides the following behaviors:

**accessIsNonSecure()**

```
slave behavior accessIsNonSecure(void) : bool;
```

**accessIsPriv()**

```
slave behavior accessIsPriv(void) : bool;
```

**addCoproprocessor()**

```
slave behavior addCoproprocessor(Coproprocessor*, int num) : void;
```

**removeCoproprocessor()**

```
slave behavior removeCoproprocessor(Coproprocessor*, int num) : void;
```

## 2.75 SystemCPChannel protocol

Defined in `$PVLIB_HOME/examples/SystemCExport/Common/Protocols/LISA/SystemCPChannelProtocol.lisa`.

### About SystemCPChannel protocol

Protocol used to communicate power state changes between a power controller and a device.

The `sg::PChannel::presp_t` enumeration provides two values:

- `sg::PChannel::ACCEPT`
- `sg::PChannel::DENY`

`SystemCPChannel` provides the following behaviors:

**pactive()**

```
master behavior pactive (uint32_t pstate) : void;
```

To be implemented by a power controller.

**prequest()**

```
slave behavior prequest (uint32_t pstate) : pchannel::presp_t;
```

To be implemented by a device.

## 2.76 SystemCoherencyInterface protocol

Defined in \$PVLIB\_HOME/LISA/SystemCoherencyInterface.lisa.

SystemCoherencyInterface provides the following behaviors:

**doDownstreamAction()**

```
optional slave behavior doDownstreamAction(const  
    SystemCoherency::DownstreamAction&) : bool;
```

**doUpstreamAction()**

```
optional master behavior doUpstreamAction(const SystemCoherency::UpstreamAction&) :  
    bool;
```

## 2.77 TZFilterControl protocol

Defined in \$PVLIB\_HOME/LISA/TZFilterUnit.lisa.

### About TZFilterControl protocol

This protocol controls the communication between filter units and control registers in the APB control block.

TZFilterControl provides the following behaviors:

**checkPermission()**

```
optional slave behavior checkPermission(const pv::TransactionAttributes*  
    attributes_,  
                                         pv::bus_addr_t page_base_,  
                                         bool is_read_,  
                                         pv::RemapRequest& req_,  
                                         bool & abort_on_error_) : bool;
```

Check the permission of the transactions filtered by the filter unit, using the information in the APB control block.

### **isEnabled()**

```
slave behavior isEnabled() : bool;
```

Check if the filter unit is enabled or not. The APB control block controls the unit.

### **isSecureSlave()**

```
optional slave behavior isSecureSlave() : bool;
```

Check if the connected slave is secure or not.

### **setConfig()**

```
optional master behavior setConfig(bool rd_spec_enable, bool wr_spec_enable,  
uint32_t action) : void;
```

Pass the configurations to the filter.

## 2.78 TZSwitchControl protocol

Defined in \$PVLIB\_HOME/LISA/TZSwitch.lisa.

### About TZSwitchControl protocol

Allow secure and normal TrustZone bus signals to be routed separately.

Transactions received on the TZSwitch `pbus_input` slave port are routed according to a configuration that is set up using parameters and/or the control port. Separate rules can be given for secure and for normal transactions.

Transactions can be routed to one of the two master ports, `pbus_port_a` or `pbus_port_b`, can be ignored, or can generate aborts.

TZSwitchControl provides the following behaviors:

### **routeAccesses()**

```
slave behavior routeAccesses(TZSwitch InputFilter input,  
TZSwitch_RouteOption destination) : void;
```

This behavior takes two arguments:

- `input` selects which types of signals are reconfigured:

#### **TZINPUT\_SECURE**

Change the routing for secure transactions

**TZINPUT\_NORMAL**

Change the routing for normal transactions

**TZINPUT\_ANY**

Change the routing for all transactions

- `destination` selects how the chosen transactions are routed:

**TZROUTE\_IGNORE**

Transactions are ignored. Reads return 0.

**TZROUTE\_TO\_PORT\_A**

Route transactions to `pvbust_port_a`.

**TZROUTE\_TO\_PORT\_B**

Route transactions to `pvbust_port_b`.

**TZROUTE\_ABORT**

Cause transactions to generate an abort.

Initial routing is configured using TZSwitch parameters `secure` and `normal` based on the following values:

<b>0</b>	Ignore
<b>1</b>	Port A
<b>2</b>	Port B
<b>3</b>	Abort

Both default and explicit parameter values are overridden by any runtime calls to `routeAccesses()` on the control port.

## 2.79 TimerCallback protocol

Defined in `$PVLIB_HOME/LISA/TimerCallbackProtocol.lisa`.

### About TimerCallback protocol

When a `ClockTimer` reaches zero, it invokes the `signal()` behavior on its `timer_callback` port. This allows a component to process a timed callback by implementing this behavior on a slave port. The slave can also return a non-zero value to retrigger the timer.

`TimerCallback` provides the following behaviors:

**signal()**

```
peer behavior signal() : uint32_t;
```

Invoked when a clock timer reaches zero. If a non-zero value is returned, the clock restarts its countdown from the returned value.

## 2.80 TimerCallback64 protocol

Defined in `$PVLIB_HOME/LISA/TimerCallbackProtocol64.lisa`.

**About TimerCallback64 protocol**

When a ClockTimer reaches zero, it invokes the `signal()` behavior on its `timer_callback` port. This allows a component to process a timed callback by implementing this behavior on a slave port. The slave can also return a non-zero value to retrigger the timer.

TimerCallback64 provides the following behaviors:

**signal()**

```
peer behavior signal() : uint64_t;
```

Invoked when a clock timer reaches zero. If a non-zero value is returned, the clock restarts its countdown from the returned value.

## 2.81 TimerControl protocol

Defined in `$PVLIB_HOME/LISA/TimerControlProtocol.lisa`.

**About TimerControl protocol**

This protocol controls the actions of the component. It permits a timer to be set to schedule a callback after a given number of ticks at the rate of the clock input.

If a timer is set while it is counting, it starts counting the new number of ticks without sending the original callback. Canceling a timer when it is not active has no effect.

TimerControl provides the following behaviors:

**cancel()**

```
slave behavior cancel() : void;
```

Cancel the countdown on the active ClockTimer, preventing the callback from being invoked.

**isSet()**

```
slave behavior isSet() : bool;
```

Test whether the timer is currently actively counting.

**remaining()**

```
slave behavior remaining() : uint32_t;
```

Return how many ticks remain before the timer's callback event will be signalled.

**set()**

```
slave behavior set(uint32_t ticks) : void;
```

Start the timer counting for the given number of ticks of its input clock. When the timer reaches zero, the scheduler invokes the `signal()` behaviour on its callback port, see [TimerCallback protocol](#).

## 2.82 TimerControl64 protocol

Defined in `$PVLIB_HOME/LISA/TimerControlProtocol64.lisa`.

**About TimerControl64 protocol**

This protocol controls the actions of the component. It permits a timer to be set to schedule a callback after a given number of ticks at the rate of the clock input.

If a timer is set while it is counting, it starts counting the new number of ticks without sending the original callback. Canceling a timer when it is not active has no effect.

TimerControl64 provides the following behaviors:

**cancel()**

```
slave behavior cancel() : void;
```

Cancel the countdown on the active ClockTimer64, preventing the callback from being invoked.

**isSet()**

```
slave behavior isSet() : bool;
```

Test whether the timer is currently actively counting.

**remaining()**

```
slave behavior remaining() : uint64_t;
```



Return how many ticks remain before the timer's callback event will be signalled.

### **set()**

```
slave behavior set(uint64_t ticks) : void;
```

Start the timer counting for the given number of ticks of its input clock. When the timer reaches zero, the scheduler invokes the `signal` behaviour on its callback port, see [TimerCallback64 protocol](#).

## 2.83 VECBProtocol protocol

Defined in `$PVLIB_HOME/examples/LISA/Common/LISA/VECBProtocol.lisa`.

`VECBProtocol` provides the following behaviors:

### **read()**

```
slave behavior read(const uint8_t function, const uint16_t device, uint32_t *  
data) : bool;
```

### **write()**

```
slave behavior write(const uint8_t function, const uint16_t device, const uint32_t  
data) : bool;
```

## 2.84 VGICComponentTraceExport protocol

Defined in `$PVLIB_HOME/LISA/VGIC_Component.lisa`.

### **About VGICComponentTraceExport protocol**

This protocol is a workaround for a LISA problem where the CADI interface of the `VGIC_Component` is not exported, but we want to export the trace sources. The trace sources can be artificially exported onto another `sg::ComponentTrace` by using this interface and not exported to the `VGIC_Component` CADI interface.

To use the `VGIC_Component`'s `export_trace` port, you must set the parameter `export_trace_to_cadi` to false, otherwise the model aborts at run time.

`VGICComponentTraceExport` provides the following behaviors:

### **exportTrace()**

```
optional slave behavior exportTrace(/*sg::ComponentTrace*/void*) : void;
```

## 2.85 VGICReportingProtocol protocol

Defined in `$PVLIB_HOME/LISA/VGIC_Component.lisa`.

`VGICReportingProtocol` provides the following behaviors:

### **logErrors()**

```
optional slave behavior logErrors( const char* buffer_ ) : void;
```

### **logFatal()**

```
optional slave behavior logFatal( const char* buffer_ ) : void;
```

### **logWarnings()**

```
optional slave behavior logWarnings( const char* buffer_ ) : void;
```

### **setEnables()**

```
optional master behavior setEnables( uint32_t new_enable_ ) : uint32_t;
```

Enable outputs on the behaviours above, returns the old value.

#### **bit[0]**

Log warnings enabled.

#### **bit[1]**

Log errors enabled.

#### **bit[2]**

Log fatal enabled.

## 2.86 Value protocol

Defined in `$PVLIB_HOME/LISA/ValueProtocol.lisa`.

### **About Value protocol**

The Value protocol allows a master to send a 32-bit unsigned value to a slave.

`value` provides the following behaviors:

### **setValue()**

```
optional peer behavior setValue(uint32_t /*value*/) : void;
```

Sets a 32-bit value for the signal.

## 2.87 ValueState protocol

Defined in `$PVLIB_HOME/LISA/ValueProtocol.lisa`.

### About ValueState protocol

The ValueState protocol allows a master to retrieve the current value from a slave.

valueState provides the following behaviors:

#### getValue()

```
peer behavior getValue() : uint32_t;
```

Returns a 32-bit value for the signal.

#### setValue()

```
peer behavior setValue(uint32_t value) : void;
```

Sets a 32-bit value for the signal.

## 2.88 ValueState\_64 protocol

Defined in `$PVLIB_HOME/LISA/Value64Protocol.lisa`.

### About ValueState\_64 protocol

The valueState\_64 protocol allows a master to retrieve the current value from a slave.

valueState\_64 provides the following behaviors:

#### getValue()

```
peer behavior getValue() : uint64_t;
```

Returns a 64-bit value for the signal.

#### setValue()

```
peer behavior setValue(uint64_t value) : void;
```

Sets a 64-bit value for the signal.

## 2.89 Value\_64 protocol

Defined in `$PVLIB_HOME/LISA/Value64Protocol.lisa`.

### About Value\_64 protocol

The `value_64` protocol allows a master to send a 64-bit unsigned value to a slave.

`value_64` provides the following behaviors:

#### **setValue()**

```
optional peer behavior setValue(uint64_t /*value*/) : void;
```

Sets a 64-bit value for the signal.

## 2.90 VirtualEthernet protocol

Defined in `$PVLIB_HOME/LISA/VirtualEthernetProtocol.lisa`.

### About VirtualEthernet protocol

The Ethernet frame class encapsulates an Ethernet frame in a broken-up format that is more accessible by components. For information on the class definition, see the `EthernetFrame.h` header file located in `$PVLIB_HOME/include/components/VirtualEthernet/Protocol/`.

`VirtualEthernet` provides the following behaviors:

#### **sendToMaster()**

```
master behavior sendToMaster(EthernetFrame* frame) : void;
```

Send an Ethernet frame to the master port.

#### **sendToSlave()**

```
slave behavior sendToSlave(EthernetFrame* frame) : void;
```

Send an Ethernet frame to the slave port.

## 2.91 VisEventRecorderProtocol protocol

Defined in `$PVLIB_HOME/LISA/VisEventRecorderProtocol.lisa`.

### About VisEventRecorderProtocol

The `VisEventRecorderProtocol` is used to play back and record events in the visualisation component of a platform system. The main purpose is for recording GUI benchmarks and

regression tests for operating systems. A master port of this protocol is in the Visualisation component and a slave port is in the VisEventRecorder component.

VisEventRecorderProtocol provides the following behaviors:

### **getEvent()**

```
slave behavior getEvent(VisEvent *event) : bool;
```

processEvents() is called to notify the master component, for example Visualisation, that new events are available.

The new events are retrieved by getEvent() from within processEvents().

The slave component decides whether playback is enabled or disabled.

Playback events:

- Return true and fill event with the next event if there is one
- Return false if there is no event

It is safe to call this behavior from outside of processEvents(). If so, it always returns false.

### **processEvents()**

```
master behavior processEvents() : void;
```

The slave component calls this behavior in the master component to notify the master component that new events are now available and must be processed.

The new events should be retrieved using getEvent() from within processEvents().

### **putEvent()**

```
slave behavior putEvent(const VisEvent *event) : void;
```

Record events call this behavior:

- Regardless of whether recording is enabled or disabled
- Even for events that just came from getEvent()

The slave component decides whether recording is enabled or disabled.

### **registerVisRegion()**

```
slave behavior registerVisRegion(VisRegion *region, const char *regionName) : void;
```

Called on initialisation. Associates names with visRegion pointers.

The slave component does not access the visRegion objects.

All `visRegion` objects, usually `visRenderRegion` and `visPushButtonRegion`, should be registered, but at least the ones where `visEvent::region` is used in the event loop.

Use the instance name for the region in the visualisation component as the name. For example `registerVisRegion(myRegion, "myRegion");`

## 2.92 v7\_VGIC\_Configuration\_Protocol protocol

Defined in `$PVLIB_HOME/LISA/v7_VGIC_Configuration_Protocol.lisa`.

`v7_VGIC_Configuration_Protocol` provides the following behaviors:

### `getNumberOfCores()`

```
slave behavior getNumberOfCores() : unsigned;
```

### `setManagerIdToCoreNumberMapping()`

```
slave behavior setManagerIdToCoreNumberMapping(
    uint64_t manager_id,
    uint64_t manager_id_mask,
    unsigned cpu_interface_number,
    unsigned inout_cluster_number,
    unsigned inout_cpu_number_in_cluster
) : bool;
```

## 2.93 v8EmbeddedCrossTrigger\_controlprotocol protocol

Defined in `$PVLIB_HOME/LISA/v8EmbeddedCrossTrigger.lisa`.

### About `v8EmbeddedCrossTrigger_controlprotocol` protocol

This protocol connects the Cross Trigger Interface (CTI) in processor components to platform-level Cross Trigger Matrix (CTM) components.

This opaque protocol is not exportable across a SystemC interface.

`v8EmbeddedCrossTrigger_controlprotocol` provides the following behaviors:

### `GetComponentIdByte()`

```
master behavior GetComponentIdByte(unsigned pidn) : uint8_t;
```

### `getPeripheralIdByte()`

```
master behavior getPeripheralIdByte(unsigned pidn) : uint8_t;
```

**getTraceInfo()**

```
master behavior getTraceInfo(SystemRegUpdateTraceProbe*& , bool*&,
    std::unordered_map<int, int>&) : void;
```

**init()**

```
slave behavior init (unsigned number_of_triggers, unsigned intack_mask, unsigned
    number_of_claim_bits, bool has_software_lock, bool has_CTIDEVCTL) : void;
```

**initDelayedSysReg()**

```
optional slave behavior initDelayedSysReg(SynchronizeSysRegHelper*, bool, bool ) :
    void;
```

**isOSUnlockCatchEnabled()**

```
optional slave behavior isOSUnlockCatchEnabled() : bool;
```

**isResetCatchEnabled()**

```
optional slave behavior isResetCatchEnabled() : bool;
```

**reg\_read()**

```
slave behavior reg_read(bool is_memory_mapped, uint32_t addr, bool is_non_secure) :
    uint32_t;
```

**reg\_write()**

```
slave behavior reg_write(bool is_memory_mapped, uint32_t addr, bool is_non_secure,
    uint32_t data) : void;
```

**reset()**

```
optional slave behavior reset() : void;
```

**setValue\_inputTrigger()**

```
slave behavior setValue_inputTrigger(unsigned index, sg::Signal::State state) :
    void;
```

**setValue\_outputTrigger()**

```
master behavior setValue_outputTrigger(unsigned index, sg::Signal::State state) :
    void;
```

## 3. Components

This chapter describes all model components in Fast Models.

For each component, the documentation includes notes about using the model, describes any deviations in the model from the Technical Reference Manual (TRM), and describes the ports and parameters.

### 3.1 Component differences

This topic lists the new and changed components in this release.

#### Differences between 11.30.27 and 11.31.15

The following components were added:

Component	Quality level
AEMvA_DSUCT	Alpha support
LabellerPAS	N/A
SDC600	N/A
SI_L1	Preliminary support
Ufs	N/A

The following components were removed:

Component
TC25_SecureAccessConfig
UFS

The following components were changed:

Component	Has the IP revision changed?	Has the quality level changed?	Have ports been added or removed?	Have parameters been added or removed?
AEMv8RMPCT	No	No	No	Yes
AEMvACT	No	No	No	Yes
AMBAPV2PVBus	No	No	No	Yes
AMBAPVACE2PVBus	No	No	No	Yes
ARMC1NanoCT	No	No	No	Yes
ARMC1NanoCT_C1ProCT	No	No	No	Yes
ARMC1NanoCT_C1ProCT_C1UltraCT	No	No	No	Yes
ARMC1NanoCT_C1UltraCT	No	No	No	Yes
ARMC1PremiumCT	No	No	No	Yes
ARMC1ProCT	No	No	No	Yes



Component	Has the IP revision changed?	Has the quality level changed?	Have ports been added or removed?	Have parameters been added or removed?
ARMC1ProCT_C1UltraCT	No	No	No	Yes
ARMC1UltraCT	No	No	No	Yes
ARMCortexA320CT	No	Yes	No	No
ARMCortexA32CT	No	No	No	Yes
ARMCortexA34CT	No	No	No	Yes
ARMCortexA35CT	No	No	No	Yes
ARMCortexA510CT	No	No	No	Yes
ARMCortexA510CT_CortexA710CT	No	No	No	Yes
ARMCortexA510CT_CortexA710CT_CortexX2CT	No	No	No	Yes
ARMCortexA510CT_CortexA710CT_CortexX3CT	No	No	No	Yes
ARMCortexA510CT_CortexA715CT_CortexX3CT	No	No	No	Yes
ARMCortexA520AECT	No	No	No	Yes
ARMCortexA520CT	No	No	No	Yes
ARMCortexA520CT_CortexA720CT	No	No	No	Yes
ARMCortexA520CT_CortexA720CT_CortexX4CT	No	No	No	Yes
ARMCortexA520CT_CortexA725CT	No	No	No	Yes
ARMCortexA520CT_CortexA725CT_CortexX925CT	No	No	No	Yes
ARMCortexA53CT	No	No	No	Yes
ARMCortexA55CT	No	No	No	Yes
ARMCortexA55CT_CortexA75CT	No	No	No	Yes
ARMCortexA55CT_CortexA76CT	No	No	No	Yes
ARMCortexA55CT_CortexA78CT	No	No	No	Yes
ARMCortexA57CT	No	No	No	Yes
ARMCortexA65AECT	No	No	No	Yes
ARMCortexA65AECT_CortexA76AECT	No	No	No	Yes
ARMCortexA65CT	No	No	No	Yes
ARMCortexA710CT	No	No	No	Yes
ARMCortexA715CT	No	No	No	Yes
ARMCortexA720AECT	No	No	No	Yes
ARMCortexA720CT	No	No	No	Yes
ARMCortexA725CT	No	No	No	Yes
ARMCortexA725CT_CortexX925CT	No	No	No	Yes
ARMCortexA72CT	No	No	No	Yes
ARMCortexA73CT	No	No	No	Yes
ARMCortexA75CT	No	No	No	Yes
ARMCortexA76AECT	No	No	No	Yes
ARMCortexA76CT	No	No	No	Yes
ARMCortexA77CT	No	No	No	Yes
ARMCortexA78AECT	No	No	No	Yes

Component	Has the IP revision changed?	Has the quality level changed?	Have ports been added or removed?	Have parameters been added or removed?
ARMCortexA78CCT	No	No	No	Yes
ARMCortexA78CT	No	No	No	Yes
ARMCortexM0CT	No	No	Yes	No
ARMCortexM0PlusCT	No	No	Yes	No
ARMCortexM23CT	No	No	Yes	No
ARMCortexM33CT	No	No	Yes	No
ARMCortexM35PCT	No	No	Yes	No
ARMCortexM52CT	No	No	Yes	No
ARMCortexM55CT	No	No	Yes	No
ARMCortexM7CT	No	No	Yes	No
ARMCortexM85CT	No	No	Yes	No
ARMCortexR52CT	No	No	No	Yes
ARMCortexR52PlusCT	No	No	No	Yes
ARMCortexR82AECT	No	No	No	Yes
ARMCortexR82CT	No	No	No	Yes
ARMCortexX1CCT	No	No	No	Yes
ARMCortexX1CT	No	No	No	Yes
ARMCortexX2CT	No	No	No	Yes
ARMCortexX3CT	No	No	No	Yes
ARMCortexX4CT	No	No	No	Yes
ARMCortexX925CT	No	No	No	Yes
ARMNeoverseE1CT	No	No	No	Yes
ARMNeoverseN1CT	No	No	No	Yes
ARMNeoverseN2CT	No	No	No	Yes
ARMNeoverseN3CT	No	No	No	Yes
ARMNeoverseV1CT	No	No	No	Yes
ARMNeoverseV2CT	No	No	No	Yes
ARMNeoverseV3AECT	No	No	No	Yes
ARMNeoverseV3CT	No	No	No	Yes
ARMSC000CT	No	No	Yes	No
AddressTranslationUnit	No	No	No	Yes
CMN700	No	No	Yes	No
CMN_S3	No	No	Yes	No
CombinedMessagingUnit	No	No	No	Yes
D71	No	No	No	Yes
ElfLoader	No	No	No	Yes
FlashLoader	No	No	No	Yes
GIC720AE	Yes	Yes	No	No
ILCU	No	No	No	Yes

Component	Has the IP revision changed?	Has the quality level changed?	Have ports been added or removed?	Have parameters been added or removed?
IntelStrataFlashJ3	No	No	No	Yes
LifeCycleManager	No	No	No	Yes
MMU_720AE	No	Yes	No	No
Mali_C720AE	No	Yes	No	No
MessageHandlingUnit	No	No	No	Yes
MessageHandlingUnitV3	No	No	No	Yes
PPUv1	No	No	Yes	No
PVBus2AMBAPV	No	No	No	Yes
PVBus2AMBAPVACE	No	No	No	Yes
PVBusBridge	No	No	No	Yes
RSE_SystemControl	No	No	Yes	No
SMMUv3AEM	No	No	No	Yes
SP810_SysCtrl	No	No	No	Yes
SecureAlarmManager	No	No	No	Yes
TrustedRAM	No	No	Yes	No
VirtioBlockDeviceMMIO	No	No	No	Yes
VirtioNetMMIO	No	No	No	Yes

## 3.2 AEMv8RMPCT

Defined in `LISA/AEMv8RMPCT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following parameters were added:

- `ADFSR-AIFSR-implemented`
- `AIDR`
- `AMIIDR`
- `AMPIDR`
- `BPIMVA_causes_translation_lookup`
- `BROADCASTATOMIC`
- `BROADCASTATOMICL`

- CCSIDR-L1D\_override
- CCSIDR-L1I\_override
- CCSIDR-L2\_override
- CCSIDR-L3\_override
- CFGTFPEN\_pin\_reset
- CHI
- CMO\_broadcast\_when\_cache\_state\_modelling\_disabled
- CPUCFR
- CTIPIDR
- CTR-L1Ip-override
- DBGBCR\_BT\_applies\_RES0\_before\_valid\_check
- DBGPIDR
- ERRIIDR
- ERRPIDR
- ERXMISC0\_mask
- MIDR
- PMCEID0
- PMCEID1
- PMSIDR.ArchInst
- PMSIDR.CRR
- PMSIDR.LDS
- PMUPIDR
- abort\_execution\_from\_device\_memory
- advsimd\_overread
- align\_pc\_on\_branch\_to\_unaligned\_pc\_aarch32
- align\_pc\_on\_debug\_exit\_to\_aarch32
- align\_pc\_on\_illegal\_exception\_return\_to\_aarch32
- amu\_aux\_type\_fixed
- amu\_mmap\_address
- amu\_num\_auxiliary\_counters
- apsr\_read\_restrict
- atomic\_memtype\_fault\_prio\_less\_than\_gpc\_fault
- atomic\_memtype\_fault\_priority
- branch-predictor-clear-policy

- branch-predictor-supported-ops
- bus\_protection\_enable\_at\_reset
- cache-log2linelen
- cache\_maintenance\_hits\_watchpoints
- changing\_block\_size\_without\_bbm\_support
- check\_memory\_attributes
- clean\_invalidate\_cache\_on\_warm\_reset
- clear\_reg\_top\_eret
- clear\_reg\_top\_set
- cluster\_utid
- configure\_pmu\_events\_with\_json
- configure\_v8\_6\_pmu\_events\_with\_json
- configure\_v8\_8\_pmu\_events\_with\_json
- configure\_v8\_9\_pmu\_events\_with\_json
- core\_cache\_protection
- cpacr\_trcds\_behaviour
- cpi\_div
- cpi\_mul
- cpu0.CP15SDISABLE2
- cpu0.CRYPTODISABLE
- cpu0.DCZID-log2-block-size
- cpu0.DCZVA\_single\_write
- cpu0.RVBAR32
- cpu0.aarch32\_reset\_from\_impdef\_addr
- cpu0.clock\_divider
- cpu0.clock\_multiplier
- cpu0.crypto\_aes
- cpu0.crypto\_sha1
- cpu0.crypto\_sha256
- cpu0.cti-intack\_mask
- cpu0.cti-number\_of\_claim\_bits
- cpu0.cti-number\_of\_triggers
- cpu0.enable\_crc32
- cpu0.enable\_trace\_special\_hlt\_imm16

- `cpu0.flash.enable`
- `cpu0.force-fpsid`
- `cpu0.force-fpsid-value`
- `cpu0.has_hcptr_tase`
- `cpu0.highest-index-of-context-breakpoints`
- `cpu0.max_code_cache_mb`
- `cpu0.number-of-breakpoints`
- `cpu0.number-of-context-breakpoints`
- `cpu0.number-of-watchpoints`
- `cpu0.operation_bandwidth`
- `cpu0.semihosting-cwd`
- `cpu0.semihosting-prefix`
- `cpu0.semihosting-stderr_istty`
- `cpu0.semihosting-stdin_istty`
- `cpu0.semihosting-stdout_istty`
- `cpu0.semihosting-use_stderr`
- `cpu0.tcm-present`
- `cpu0.tcm-supports-exclusive`
- `cpu0.tcm.a.base`
- `cpu0.tcm.a.enable`
- `cpu0.tcm.a.size`
- `cpu0.tcm.a.stretch_clk`
- `cpu0.tcm.a.wait`
- `cpu0.tcm.b.base`
- `cpu0.tcm.b.enable`
- `cpu0.tcm.b.size`
- `cpu0.tcm.b.stretch_clk`
- `cpu0.tcm.b.wait`
- `cpu0.tcm.c.base`
- `cpu0.tcm.c.enable`
- `cpu0.tcm.c.size`
- `cpu0.tcm.c.stretch_clk`
- `cpu0.tcm.c.wait`
- `cpu0.trace_special_hlt_imm16`

- `cpu0.unpredictable_WPMASKANDBAS`
- `cpu0.vfp-dp-present`
- `cpu0.vfp-traps`
- `cpu0.vfp-traps-show-all`
- `cpu0.wfet_early_or_delayed_timeout`
- `cpu0.wfit_early_or_delayed_timeout`
- `cpu1.CP15SDISABLE2`
- `cpu1.CRYPTODISABLE`
- `cpu1.DCZID-log2-block-size`
- `cpu1.DCZVA_single_write`
- `cpu1.RVBAR32`
- `cpu1.aarch32_reset_from_impdef_addr`
- `cpu1.clock_divider`
- `cpu1.clock_multiplier`
- `cpu1.crypto_aes`
- `cpu1.crypto_shal`
- `cpu1.crypto_sha256`
- `cpu1.cti-intack_mask`
- `cpu1.cti-number_of_claim_bits`
- `cpu1.cti-number_of_triggers`
- `cpu1.enable_crc32`
- `cpu1.enable_trace_special_hlt_imm16`
- `cpu1.flash.enable`
- `cpu1.force-fpsid`
- `cpu1.force-fpsid-value`
- `cpu1.has_hcptr_tase`
- `cpu1.highest-index-of-context-breakpoints`
- `cpu1.max_code_cache_mb`
- `cpu1.number-of-breakpoints`
- `cpu1.number-of-context-breakpoints`
- `cpu1.number-of-watchpoints`
- `cpu1.operation_bandwidth`
- `cpu1.semihosting-cwd`
- `cpu1.semihosting-prefix`

- `cpu1.semihosting-stderr_istty`
- `cpu1.semihosting-stdin_istty`
- `cpu1.semihosting-stdout_istty`
- `cpu1.semihosting-use_stderr`
- `cpu1.tcm-present`
- `cpu1.tcm-supports-exclusive`
- `cpu1.tcm.a.base`
- `cpu1.tcm.a.enable`
- `cpu1.tcm.a.size`
- `cpu1.tcm.a.stretch_clk`
- `cpu1.tcm.a.wait`
- `cpu1.tcm.b.base`
- `cpu1.tcm.b.enable`
- `cpu1.tcm.b.size`
- `cpu1.tcm.b.stretch_clk`
- `cpu1.tcm.b.wait`
- `cpu1.tcm.c.base`
- `cpu1.tcm.c.enable`
- `cpu1.tcm.c.size`
- `cpu1.tcm.c.stretch_clk`
- `cpu1.tcm.c.wait`
- `cpu1.trace_special_hlt_imm16`
- `cpu1.unpredictable_WPMASKANDBAS`
- `cpu1.vfp-dp-present`
- `cpu1.vfp-traps`
- `cpu1.vfp-traps-show-all`
- `cpu1.wfet_early_or_delayed_timeout`
- `cpu1.wfit_early_or_delayed_timeout`
- `cpu2.CP15SDISABLE2`
- `cpu2.CRYPTODISABLE`
- `cpu2.DCZID-log2-block-size`
- `cpu2.DCZVA_single_write`
- `cpu2.RVBAR32`
- `cpu2.aarch32_reset_from_impdef_addr`



- `cpu2.clock_divider`
- `cpu2.clock_multiplier`
- `cpu2.crypto_aes`
- `cpu2.crypto_shal`
- `cpu2.crypto_sha256`
- `cpu2.cti-intack_mask`
- `cpu2.cti-number_of_claim_bits`
- `cpu2.cti-number_of_triggers`
- `cpu2.enable_crc32`
- `cpu2.enable_trace_special_hlt_imm16`
- `cpu2.flash.enable`
- `cpu2.force-fpsid`
- `cpu2.force-fpsid-value`
- `cpu2.has_hcptr_tase`
- `cpu2.highest-index-of-context-breakpoints`
- `cpu2.max_code_cache_mb`
- `cpu2.number-of-breakpoints`
- `cpu2.number-of-context-breakpoints`
- `cpu2.number-of-watchpoints`
- `cpu2.operation_bandwidth`
- `cpu2.semihosting-cwd`
- `cpu2.semihosting-prefix`
- `cpu2.semihosting-stderr_istty`
- `cpu2.semihosting-stdin_istty`
- `cpu2.semihosting-stdout_istty`
- `cpu2.semihosting-use_stderr`
- `cpu2.tcm-present`
- `cpu2.tcm-supports-exclusive`
- `cpu2.tcm.a.base`
- `cpu2.tcm.a.enable`
- `cpu2.tcm.a.size`
- `cpu2.tcm.a.stretch_clk`
- `cpu2.tcm.a.wait`
- `cpu2.tcm.b.base`

- `cpu2.tcm.b.enable`
- `cpu2.tcm.b.size`
- `cpu2.tcm.b.stretch_clk`
- `cpu2.tcm.b.wait`
- `cpu2.tcm.c.base`
- `cpu2.tcm.c.enable`
- `cpu2.tcm.c.size`
- `cpu2.tcm.c.stretch_clk`
- `cpu2.tcm.c.wait`
- `cpu2.trace_special_hlt_imm16`
- `cpu2.unpredictable_WPMASKANDBAS`
- `cpu2.vfp-dp-present`
- `cpu2.vfp-traps`
- `cpu2.vfp-traps-show-all`
- `cpu2.wfet_early_or_delayed_timeout`
- `cpu2.wfit_early_or_delayed_timeout`
- `cpu3.CP15SDISABLE2`
- `cpu3.CRYPTODISABLE`
- `cpu3.DCZID-log2-block-size`
- `cpu3.DCZVA_single_write`
- `cpu3.RVBAR32`
- `cpu3.aarch32_reset_from_impdef_addr`
- `cpu3.clock_divider`
- `cpu3.clock_multiplier`
- `cpu3.crypto_aes`
- `cpu3.crypto_shal`
- `cpu3.crypto_sha256`
- `cpu3.cti-intack_mask`
- `cpu3.cti-number_of_claim_bits`
- `cpu3.cti-number_of_triggers`
- `cpu3.enable_crc32`
- `cpu3.enable_trace_special_hlt_imm16`
- `cpu3.flash.enable`
- `cpu3.force-fpsid`

- `cpu3.force-fpsid-value`
- `cpu3.has_hcptr_tase`
- `cpu3.highest-index-of-context-breakpoints`
- `cpu3.max_code_cache_mb`
- `cpu3.number-of-breakpoints`
- `cpu3.number-of-context-breakpoints`
- `cpu3.number-of-watchpoints`
- `cpu3.operation_bandwidth`
- `cpu3.semihosting-cwd`
- `cpu3.semihosting-prefix`
- `cpu3.semihosting-stderr_istty`
- `cpu3.semihosting-stdin_istty`
- `cpu3.semihosting-stdout_istty`
- `cpu3.semihosting-use_stderr`
- `cpu3.tcm-present`
- `cpu3.tcm-supports-exclusive`
- `cpu3.tcm.a.base`
- `cpu3.tcm.a.enable`
- `cpu3.tcm.a.size`
- `cpu3.tcm.a.stretch_clk`
- `cpu3.tcm.a.wait`
- `cpu3.tcm.b.base`
- `cpu3.tcm.b.enable`
- `cpu3.tcm.b.size`
- `cpu3.tcm.b.stretch_clk`
- `cpu3.tcm.b.wait`
- `cpu3.tcm.c.base`
- `cpu3.tcm.c.enable`
- `cpu3.tcm.c.size`
- `cpu3.tcm.c.stretch_clk`
- `cpu3.tcm.c.wait`
- `cpu3.trace_special_hlt_imm16`
- `cpu3.unpredictable_WPMASKANDBAS`
- `cpu3.vfp-dp-present`

- `cpu3.vfp-traps`
- `cpu3.vfp-traps-show-all`
- `cpu3.wfet_early_or_delayed_timeout`
- `cpu3.wfit_early_or_delayed_timeout`
- `dbg-bcr-reserved-behavior`
- `dbg_rom_dap_addr`
- `dbgitr_buffer_size`
- `dbgxvr_ress_is_stateful`
- `dc_fault_unaligned_s1_device_s2_fwb`
- `dcache-hit_latency`
- `dcache-maintenance_latency`
- `dcache-miss_latency`
- `dcache-prefetch_enabled`
- `dcache-read_access_latency`
- `dcache-read_bus_width_in_bytes`
- `dcache-read_latency`
- `dcache-size`
- `dcache-snoop_data_transfer_latency`
- `dcache-ways`
- `dcache-write_access_latency`
- `dcache-write_bus_width_in_bytes`
- `dcache-write_latency`
- `dcimva_requires_s2_write_permissions`
- `debug_auth_signals_sampled_at_reset`
- `debug_components_dap_address`
- `debug_components_mmap_address`
- `debug_entry_is_context_sync`
- `debug_rom_is_class_9`
- `debug_rom_is_flat`
- `def_mem_map`
- `def_mem_map_file_path`
- `default_inner_shareable`
- `delay_serror`
- `dic-spi_count`

- `disable_impdef_abort_on_ic_maintenance`
- `disable_unknown_update_event_on_reset`
- `dsb_accumulate_threshold`
- `edpfr_ras_unknown_bits_read_as_0`
- `el0_can_access_imp_def_functionality`
- `el3_trap_priority_when_secure_debug_disabled`
- `enable-gicv5`
- `enable_address_contig_check`
- `enable_debug_auth_signals_config`
- `enable_lock_step`
- `error_record_feature_register_json_file`
- `erxpfgctl_res0_stateful_mask`
- `exception_catch_before_software_step`
- `exception_catch_type`
- `exclusive_monitor_clear_on_atomic_from_same_master`
- `exclusive_monitor_clear_on_store_from_same_master`
- `exclusive_monitor_clear_on_strex_address_mismatch`
- `exclusive_monitor_clear_on_strex_success`
- `exercise_stxr_fail`
- `ext_abort_device_GRE_prefetch_ras_index`
- `ext_abort_device_GRE_prefetch_ras_type`
- `ext_abort_device_GRE_read_is_critical`
- `ext_abort_device_GRE_read_is_sync`
- `ext_abort_device_GRE_read_ras_index`
- `ext_abort_device_GRE_read_ras_type`
- `ext_abort_device_GRE_write_is_critical`
- `ext_abort_device_GRE_write_is_sync`
- `ext_abort_device_GRE_write_ras_index`
- `ext_abort_device_GRE_write_ras_type`
- `ext_abort_device_nGRE_prefetch_ras_index`
- `ext_abort_device_nGRE_prefetch_ras_type`
- `ext_abort_device_nGRE_read_is_critical`
- `ext_abort_device_nGRE_read_is_sync`
- `ext_abort_device_nGRE_read_ras_index`

- ext\_abort\_device\_nGRE\_read\_ras\_type
- ext\_abort\_device\_nGRE\_write\_is\_critical
- ext\_abort\_device\_nGRE\_write\_is\_sync
- ext\_abort\_device\_nGRE\_write\_ras\_index
- ext\_abort\_device\_nGRE\_write\_ras\_type
- ext\_abort\_device\_prefetch\_ras\_index
- ext\_abort\_device\_prefetch\_ras\_type
- ext\_abort\_device\_read\_acquire\_is\_sync
- ext\_abort\_device\_read\_is\_critical
- ext\_abort\_device\_read\_is\_sync
- ext\_abort\_device\_read\_ras\_index
- ext\_abort\_device\_read\_ras\_type
- ext\_abort\_device\_write\_is\_critical
- ext\_abort\_device\_write\_is\_sync
- ext\_abort\_device\_write\_ras\_index
- ext\_abort\_device\_write\_ras\_type
- ext\_abort\_fill\_data
- ext\_abort\_in\_virtual\_tag\_reads\_is\_sync
- ext\_abort\_in\_virtual\_tag\_writes\_is\_sync
- ext\_abort\_normal\_cacheable\_read\_is\_critical
- ext\_abort\_normal\_cacheable\_read\_is\_sync
- ext\_abort\_normal\_cacheable\_read\_ras\_index
- ext\_abort\_normal\_cacheable\_read\_ras\_type
- ext\_abort\_normal\_cacheable\_write\_is\_critical
- ext\_abort\_normal\_cacheable\_write\_is\_sync
- ext\_abort\_normal\_cacheable\_write\_ras\_index
- ext\_abort\_normal\_cacheable\_write\_ras\_type
- ext\_abort\_normal\_noncacheable\_prefetch\_ras\_index
- ext\_abort\_normal\_noncacheable\_prefetch\_ras\_type
- ext\_abort\_normal\_noncacheable\_read\_is\_critical
- ext\_abort\_normal\_noncacheable\_read\_is\_sync
- ext\_abort\_normal\_noncacheable\_read\_ras\_index
- ext\_abort\_normal\_noncacheable\_read\_ras\_type
- ext\_abort\_normal\_noncacheable\_write\_is\_critical

- `ext_abort_normal_noncacheable_write_is_sync`
- `ext_abort_normal_noncacheable_write_ras_index`
- `ext_abort_normal_noncacheable_write_ras_type`
- `ext_abort_normal_wt_cacheable_prefetch_ras_index`
- `ext_abort_normal_wt_cacheable_prefetch_ras_type`
- `ext_abort_normal_wt_cacheable_read_is_critical`
- `ext_abort_normal_wt_cacheable_read_is_sync`
- `ext_abort_normal_wt_cacheable_read_ras_index`
- `ext_abort_normal_wt_cacheable_read_ras_type`
- `ext_abort_normal_wt_cacheable_write_is_critical`
- `ext_abort_normal_wt_cacheable_write_is_sync`
- `ext_abort_normal_wt_cacheable_write_ras_index`
- `ext_abort_normal_wt_cacheable_write_ras_type`
- `ext_abort_prefetch_device_GRE_read_is_critical`
- `ext_abort_prefetch_device_GRE_read_is_sync`
- `ext_abort_prefetch_device_nGRE_read_is_critical`
- `ext_abort_prefetch_device_nGRE_read_is_sync`
- `ext_abort_prefetch_device_read_is_critical`
- `ext_abort_prefetch_device_read_is_sync`
- `ext_abort_prefetch_is_critical`
- `ext_abort_prefetch_is_sync`
- `ext_abort_prefetch_noncacheable_read_is_critical`
- `ext_abort_prefetch_noncacheable_read_is_sync`
- `ext_abort_prefetch_ras_index`
- `ext_abort_prefetch_ras_type`
- `ext_abort_prefetch_so_read_is_critical`
- `ext_abort_prefetch_so_read_is_sync`
- `ext_abort_prefetch_wt_cacheable_read_is_critical`
- `ext_abort_prefetch_wt_cacheable_read_is_sync`
- `ext_abort_so_prefetch_ras_index`
- `ext_abort_so_prefetch_ras_type`
- `ext_abort_so_read_is_critical`
- `ext_abort_so_read_is_sync`
- `ext_abort_so_read_ras_index`

- `ext_abort_so_read_ras_type`
- `ext_abort_so_write_is_critical`
- `ext_abort_so_write_is_sync`
- `ext_abort_so_write_ras_index`
- `ext_abort_so_write_ras_type`
- `ext_abort_ttw_cacheable_read_is_critical`
- `ext_abort_ttw_cacheable_read_is_sync`
- `ext_abort_ttw_cacheable_read_ras_index`
- `ext_abort_ttw_cacheable_read_ras_type`
- `ext_abort_ttw_noncacheable_read_is_critical`
- `ext_abort_ttw_noncacheable_read_is_sync`
- `ext_abort_ttw_noncacheable_read_ras_index`
- `ext_abort_ttw_noncacheable_read_ras_type`
- `ext_abort_ttw_wt_cacheable_read_is_critical`
- `ext_abort_ttw_wt_cacheable_read_is_sync`
- `ext_abort_ttw_wt_cacheable_read_ras_index`
- `ext_abort_ttw_wt_cacheable_read_ras_type`
- `external_oslar_access_disabled_by_authentication`
- `far_unchanged_for_stack_alignment_fault`
- `far_unchanged_for_virtual_serror`
- `far_unchanged_when_fnv_set`
- `fault_on_nT_bit_set`
- `fault_unalign_to_unsupported_access`
- `fault_unaligned_s1_device_s2_fwb`
- `flash_protection_enable_at_reset`
- `force_align_pc`
- `force_sync_on_wfx`
- `fpcr_short_vector_raz`
- `fpsr_res0_stateful_mask`
- `gic.GICV-alias`
- `gic_iri.ARE-fixed-to-one`
- `gic_iri.DPG-ARE-only`
- `gic_iri.DPG-bits-implemented`
- `gic_iri.GICD-alias`



- `gic_iri.GICD_ITARGETSR-RAZWI`
- `gic_iri.GICD_PIDR`
- `gic_iri.GICR_PIDR`
- `gic_iri.GITS_PIDR`
- `gic_iri.ICFGR-rsvd-bit`
- `gic_iri.IIDR`
- `gic_iri.IRI-ID-bits`
- `gic_iri.ITS-count`
- `gic_iri.SPI-count`
- `gic_iri.SPI-message-based-support`
- `gic_iri.STATUSR-implemented`
- `gic_iri.enable_protocol_checking`
- `gic_iri.enabled`
- `gic_iri.has-two-security-states`
- `gic_iri.irouter-default-mask`
- `gic_iri.irouter-default-reset`
- `gic_iri.monolithic`
- `gic_iri.non-ARE-core-count`
- `gic_iri.periph-size`
- `gic_iri.priority-bits`
- `gic_iri.processor-numbers`
- `gic_iri.redistributor-offset`
- `gic_iri.redistributor-size`
- `gic_iri.reg-offset`
- `gic_iri.supports-shareability`
- `gic_iri.virtual-lpi-support`
- `gic_iri.wakeup-on-reset`
- `gicv3.A3-affinity-supported`
- `gicv3.BPR-min`
- `gicv3.EOI-check-CPUID`
- `gicv3.EOI-check-ID`
- `gicv3.EOI-deactivate-any-interrupt`
- `gicv3.EOI-ignore-out-of-order`
- `gicv3.FIQEn-RAO`

- gicv3.IIDR\_base
- gicv3.LR-count
- gicv3.PMHE-~~RAO-WI~~
- gicv3.PMHE-~~RAZ-WI~~
- gicv3.PMHE-release-set-packet
- gicv3.SRE-EL2-enable-~~RAO~~
- gicv3.SRE-EL3-enable-~~RAO~~
- gicv3.SRE-EL3-set-once
- gicv3.SRE-enable-action-on-mmap
- gicv3.VBPR-min
- gicv3.VFIQEn-~~RAO~~
- gicv3.dir-trap-support
- gicv3.el3\_trap\_priority\_when\_secure\_debug\_disabled
- gicv3.extended-interrupt-range-support
- gicv3.idle-is-ff
- gicv3.ignore-DIR-write-when-EOImode-not-set
- gicv3.interrupt-bypass-support
- gicv3.local-SEIs
- gicv3.local-VSEIs
- gicv3.physical-ID-bits
- gicv3.send-PMHE-command-only-when-priority-changes
- gicv3.sgi-range-selector-support
- gicv3.suppress-virtual-enables-comms
- gicv3.virtual-ID-bits
- gicv3.virtual-lpi-support
- gicv3.without-DS-support
- gicv4.mask-virtual-interrupt
- gicv5.config\_file
- gicv5.has\_gcie\_legacy
- gicv5.interrupt-bypass-support
- global\_debug\_rom.ROMDEVID
- global\_debug\_rom.ROMPIDR
- global\_debug\_rom.ROMPRIDR0
- has\_16bit\_asids

- `has_16k_granule`
- `has_4k_granule`
- `has_64k_granule`
- `has_aarch32_dbgdidr_etc`
- `has_bc`
- `has_cluster_llcache_size`
- `has_coherent_icache`
- `has_const_pac`
- `has_cvadp_support`
- `has_debug_rom`
- `has_delayed_ctireg`
- `has_delayed_dbgreg`
- `has_delayed_mdscr_el1`
- `has_delayed_oslar_el1`
- `has_delayed_pmureg`
- `has_delayed_sysreg`
- `has_dgh`
- `has_edacr`
- `has_exception_trapping_form_of_vector_catch`
- `has_export_m_port`
- `has_far_not_valid`
- `has_far_not_valid_dfsc`
- `has_far_not_valid_ifsc`
- `has_flash_protection`
- `has_generic_authentication`
- `has_hardware_translation_table_update`
- `has_internal_gic_iri`
- `has_itd`
- `has_large_system_ext`
- `has_large_va`
- `has_llpp`
- `has_mpm`
- `has_par_bit10_razwi`
- `has_partial_delayed_mdscr_el1`

- `has_pc_sample_based_profiling`
- `has_per_cluster_debug_auth_ports`
- `has_pmc`
- `has_pmu`
- `has_pointer_authentication`
- `has_qarma3_pac`
- `has_ras`
- `has_ras_armv84_extension`
- `has_ras_double_fault`
- `has_small_page_table`
- `has_software_lock`
- `has_speculative_sei`
- `has_spp`
- `has_stage2_ap_speculative_update`
- `has_synchronous_load_atomics`
- `has_synchronous_load_atomics_noncacheable`
- `has_synchronous_store_atomics`
- `has_synchronous_store_atomics_noncacheable`
- `has_tlb_conflict_abort`
- `has_tlb_pa_caching`
- `has_unsupported_exclusive_fault`
- `has_v8_4_pmu_extension`
- `has_v8_6_pmu_events`
- `has_writebuffer`
- `hcptr_tta_behaviour`
- `hcr_el2_miocnce_is_rw`
- `hcr_swio_res1`
- `hsr_uncond_cc`
- `icache-hit_latency`
- `icache-log2linelen`
- `icache-maintenance_latency`
- `icache-miss_latency`
- `icache-nprefetch`
- `icache-prefetch_enabled`

- `icache-prefetch_level`
- `icache-read_access_latency`
- `icache-read_bus_width_in_bytes`
- `icache-read_latency`
- `icache-size`
- `icache-ways`
- `ignore_tag_check_dcc_load_store_in_ma_mode_when_tco_is_disabled`
- `imp_def_functionality_behaviour`
- `inner_cache_boundary`
- `instruction_tlb_size`
- `is_debug_state_pmu_snapshot_allowed`
- `is_first_pcsr_sample_ignored`
- `is_ras_irq_edge_triggered`
- `is_serror_edge_triggered`
- `is_uniprocessor`
- `isb_is_branch`
- `ish_is_osh`
- `itd_conditional_instructions_are_32bit`
- `jidr_is_undef_at_el0`
- `jmcr_is_undef_at_el0`
- `joscr_is_undef_at_el0`
- `l1cache_has_r52_cache_policy`
- `l1cache_has_rsvd_flash_ways`
- `l2cache-hit_latency`
- `l2cache-maintenance_latency`
- `l2cache-miss_latency`
- `l2cache-read_access_latency`
- `l2cache-read_bus_width_in_bytes`
- `l2cache-read_latency`
- `l2cache-size`
- `l2cache-snoop_data_transfer_latency`
- `l2cache-snoop_issue_latency`
- `l2cache-ways`
- `l2cache-write_access_latency`

- l2cache-write\_bus\_width\_in\_bytes
- l2cache-write\_latency
- l3cache-has\_mpam
- l3cache-hit\_latency
- l3cache-maintenance\_latency
- l3cache-miss\_latency
- l3cache-mpamf.arch\_major\_ver
- l3cache-mpamf.arch\_minor\_ver
- l3cache-mpamf.esr\_mask
- l3cache-mpamf.has\_esr
- l3cache-mpamf.has\_extd\_esr
- l3cache-mpamf.has\_impl\_idr
- l3cache-mpamf.has\_mbwu\_long\_counter
- l3cache-mpamf.has\_mpamfidr\_ext
- l3cache-mpamf.has\_partid\_nrw
- l3cache-mpamf.has\_priority\_partitioning
- l3cache-mpamf.has\_prod\_id
- l3cache-mpamf.has\_prod\_rev
- l3cache-mpamf.has\_prod\_var
- l3cache-mpamf.has\_ris
- l3cache-mpamf.max\_partid\_ns
- l3cache-mpamf.max\_partid\_rl
- l3cache-mpamf.max\_partid\_rt
- l3cache-mpamf.max\_partid\_s
- l3cache-mpamf.max\_pmg\_ns
- l3cache-mpamf.max\_pmg\_rl
- l3cache-mpamf.max\_pmg\_rt
- l3cache-mpamf.max\_pmg\_s
- l3cache-mpamf.mbwu\_long\_counter\_width
- l3cache-mpamf.no\_impl\_msmon
- l3cache-mpamf.no\_impl\_part
- l3cache-mpamf.ris\_max
- l3cache-mpamf\_base
- l3cache-read\_access\_latency

- l3cache-read\_bus\_width\_in\_bytes
- l3cache-read\_latency
- l3cache-size
- l3cache-snoop\_data\_transfer\_latency
- l3cache-snoop\_issue\_latency
- l3cache-ways
- l3cache-write\_access\_latency
- l3cache-write\_bus\_width\_in\_bytes
- l3cache-write\_latency
- legacy\_combining\_exc\_catch\_trace
- limit\_ete\_revision\_without\_rme
- ls64\_ignore\_sl\_unpred\_memattr\_transformation
- ls64\_memtype\_check\_use\_combined\_memattr
- ls64wb\_memtype\_check\_allow\_any\_cacheable\_memattr
- mdrar\_ell\_res0
- memory.ext\_slave\_base
- memory.ext\_slave\_size\_per\_core
- memory.flash\_base
- memory.flash\_size
- memory.has\_llram
- memory.l2\_cache.is\_inner\_cacheable
- memory.l2\_cache.is\_inner\_shareable
- memory.llram\_base
- memory.llram\_enable\_at\_reset
- memory.llram\_shared
- memory.llram\_size
- memory.scu\_present
- mixed\_endian
- mpidr\_layout
- num\_protection\_regions\_s1
- num\_protection\_regions\_s2
- num\_spi
- par\_ns\_set\_unknown\_bit
- par\_nse\_set\_unknown\_bit

- `per_core_master_supported`
- `pfr1_csv2_frac`
- `plbi_invalid_xt`
- `pmb_idr_external_abort`
- `pmb_idr_flag_updates`
- `pmbsr_dl_razwi`
- `pmbsr_ea_razwi`
- `pmcr_disable_events_export`
- `pmmir_ell_bus_slots`
- `pmmir_ell_bus_width`
- `pms_idr_max_size`
- `pmu-num_counters`
- `pmu_cycle_counter_counts_actual_cycles`
- `pmu_has_chain_event`
- `preserve_cache_contents_over_warm_reset`
- `ptw_latency`
- `ram_protection_enable_at_reset`
- `randomize_unknowns_at_reset`
- `ras_extra_configurations`
- `ras_pfg_clock_mhz`
- `register_reset_data`
- `register_reset_data_hi`
- `report_iside_cmo_ifsr`
- `report_second_access_align_fault_non_atomic_pair_access`
- `report_second_access_mmu_fault_non_atomic_pair_access`
- `reported_fp_revision`
- `reported_patch_level`
- `reported_revision_number`
- `reserved_HMC_SSC_PAC_treated_disabled`
- `restore_fpsr_on_trapped_fp_exception`
- `s1_align_memtype_fault_prio_more_than_s2_perm_fault_on_s1_walk`
- `s1_perm_fault_prio_more_than_s2_perm_fault_on_s1_walk`
- `s1_unsupported_atomic_fault_for_ls64_prio_more_than_s2_perm_fault`
- `scheduler_mode`



- `scr_nET_writeable`
- `scramble_unknowns_at_reset`
- `serror_clear_delay`
- `skip_trace_on_write_to_oseccr_el1_when_oslock_is_unlocked`
- `spp.base`
- `spp.size`
- `spsr_el3_is_mapped_to_spsr_mon`
- `spsr_m4_res0`
- `stage12_tlb_size`
- `stage1_tlb_size`
- `stage1_walkcache_size`
- `strex_fail_can_hit_watchpoint`
- `supports_multi_threading`
- `swp_with_xzr_is_st_atomic`
- `take_ccfail_tsc_trap`
- `take_ccfail_undef`
- `tcr_tgx_bit1_stateful`
- `tidcp_traps_el0_undef_imp_def`
- `tlb_latency`
- `tlbi_or_ic_invalid_xt`
- `trace_has_sysreg_access`
- `trace_icc_registers_as_icv_when_redirected`
- `trace_physical_registers_when_host_virtualisation_enabled`
- `trace_xzr_in_core_regs64_trace`
- `trap_dc_cmo_to_pou_if_nop`
- `trap_ic_cmo_to_pou_if_nop`
- `trap_reserved_group3_id_regs`
- `treat-dcache-cmos-to-poc-as-nop`
- `treat-dcache-cmos-to-pou-as-nop`
- `treat-dcache-invalidate-as-clean-invalidate`
- `treat-icache-cmos-to-pou-as-nop`
- `treat_forced_normal_as_device_for_excl_atomics`
- `treat_pld_as_nop`
- `treat_pli_as_nop`

- `treat_wfi_wfe_as_nop`
- `truncate_pc_on_illegal_exception_return_to_aarch32`
- `unification-level`
- `unification-uniprocessor-level`
- `unpred_LSE128_overlap`
- `unpred_brbe_next_branch_cycle_count_unknown`
- `unpred_clear_ISV_for_exception_before_software_step`
- `unpred_edscr_ns_set_unknown_bit`
- `unpred_edscr_rw_unknown_bits_read_as_1`
- `unpred_edscr_status_read_as_no_syndrome`
- `unpred_extdbg_unknown_bits`
- `unpred_load_single_reg_overlap_with_wb`
- `unpred_mrsmsr_currentlymapped_undef`
- `unpred_mrsmsr_protfailed_undef`
- `unpred_par_attr_returns_mair`
- `unpred_s2_hw_dirty_update_on_atomic_wo_read_perm_fault`
- `unpred_sctlr_c_0_taggable_behaviour`
- `unpred_stage2_mpu_and_bg_disabled`
- `unpred_store_exclusive_base_overlap`
- `unpred_store_pair_and_single_reg_overlap_with_wb`
- `unpred_tchange_tenter_and_texit_behaviour`
- `unpred_tlbi_not_in_monitor_mode`
- `unpred_tsize_aborts`
- `unpred_tsize_pamax_aborts`
- `unpredictable_exclusive_abort_memtype`
- `unpredictable_hvc_behaviour`
- `unpredictable_smc_behaviour`
- `unpredictable_wfet_and_wfit_behaviour`
- `unsupported_atomic_fault_type`
- `unsupported_hw_update_fault_type`
- `use_architectural_names`
- `use_stage1_sh_as_input_to_stage2`
- `use_tlb_contig_hint`
- `user_defined_rom_table_debug_power_config`

- vpu\_datapath\_width
- walk\_cache\_latency
- watchpoint-log2secondary\_restriction
- wfe\_wakeup\_delay
- wfi\_wakeup\_delay
- wp\_ignores\_dbm\_update

The following parameters were removed:

- cpu1.CONFIG64
- cpu2.CONFIG64
- cpu3.CONFIG64

## About AEMv8RMPCT

AEMv8RMPCT CPU component.

## Iris and MTI instances for AEMv8RMPCT

This model has the following Iris instances:

Name	Instance type
AEMv8RMPCT	Cluster_ARMAEMv8-R_MP
AEMv8RMPCT.AMU	PVBusLogger
AEMv8RMPCT.AMU.mapper	PVBusMapper
AEMv8RMPCT.DAP	PVBusLogger
AEMv8RMPCT.DAP.mapper	PVBusMapper
AEMv8RMPCT.MMAP	PVBusLogger
AEMv8RMPCT.MMAP.mapper	PVBusMapper
AEMv8RMPCT.RAS	PVBusLogger
AEMv8RMPCT.RAS.mapper	PVBusMapper
AEMv8RMPCT.acp_mapper	PVBusMapper
AEMv8RMPCT.cpu0	ARMAEMv8-R_MP
AEMv8RMPCT.cpu0.UTLB	TLB
AEMv8RMPCT.cpu0.debug_rom	debug_rom
AEMv8RMPCT.cpu0.dtlb	TLB
AEMv8RMPCT.cpu0.l1dcache	PVCache
AEMv8RMPCT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
AEMv8RMPCT.cpu0.l1dcache.upstream[0]	PVBusSlave
AEMv8RMPCT.cpu0.l1licache	PVCache
AEMv8RMPCT.cpu0.l1licache.downstream[0].pvbusmaster	PVBusMaster
AEMv8RMPCT.cpu0.l1licache.upstream[0]	PVBusSlave
AEMv8RMPCT.ext_bus	PVBusLogger
AEMv8RMPCT.ext_bus.mapper	PVBusMapper

Name	Instance type
AEMv8RMPCT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
AEMv8RMPCT.global_debug_rom	debug_rom
AEMv8RMPCT.l2_cache	PVCache
AEMv8RMPCT.l2_cache.downstream[0].pvbusmaster	PVBusMaster
AEMv8RMPCT.l2_cache.upstream[Z] (where Z = 0-16)	PVBusSlave
AEMv8RMPCT.l2_flusher	AsyncCacheFlushUnit
AEMv8RMPCT.secondary_debug_rom	debug_rom

This model has the following MTI trace components:

Name	Component type
AEMv8RMPCT.AMU	PVBusLogger
AEMv8RMPCT.AMU.mapper	PVBusMapper
AEMv8RMPCT.DAP	PVBusLogger
AEMv8RMPCT.DAP.mapper	PVBusMapper
AEMv8RMPCT.MMAP	PVBusLogger
AEMv8RMPCT.MMAP.mapper	PVBusMapper
AEMv8RMPCT.RAS	PVBusLogger
AEMv8RMPCT.RAS.mapper	PVBusMapper
AEMv8RMPCT.acp_mapper	PVBusMapper
AEMv8RMPCT.cpu0	ARM_AEMv8-R_MP
AEMv8RMPCT.cpu0.UTLB	TLB
AEMv8RMPCT.cpu0.l1dcache	PVCache
AEMv8RMPCT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
AEMv8RMPCT.cpu0.l1dcache.upstream[0]	PVBusSlave
AEMv8RMPCT.cpu0.l1icache	PVCache
AEMv8RMPCT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
AEMv8RMPCT.cpu0.l1icache.upstream[0]	PVBusSlave
AEMv8RMPCT.ext_bus	PVBusLogger
AEMv8RMPCT.ext_bus.mapper	PVBusMapper
AEMv8RMPCT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
AEMv8RMPCT.l2_cache	PVCache
AEMv8RMPCT.l2_cache.downstream[0].pvbusmaster	PVBusMaster
AEMv8RMPCT.l2_cache.upstream[Z] (where Z = 0-16)	PVBusSlave
AEMv8RMPCT.l2_flusher	AsyncCacheFlushUnit

## Ports for AEMv8RMPCT

Port	Direction	Protocol	Description
broadcastcachemaint	slave	Signal	Enable broadcasting of cache maintenance operations to downstream caches.
broadcastinner	slave	Signal	Enable broadcasting of Inner Shareable transactions.

Port	Direction	Protocol	Description
broadcastouter	slave	Signal	Enable broadcasting of Outer Shareable transactions.
cache_validation_control	slave	Value	This signal provides default exception handling state.
cfgend	slave	Signal	This signal is for EE bit initialisation
cfgsdisable	slave	Signal	This signal disables write access to some secure Interrupt Controller registers
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
clusterid	slave	Value	The port reads the value in CPU ID register field, bits[11:8] of the MPIDR.
CNTHPIRQ	master	Signal	Timer signals to SOC
CNTHPSIRQ	master	Signal	Timer signals to SOC.
CNTHVSIQ	master	Signal	Timer signals to SOC.
CNTPNSIRQ	master	Signal	Timer signals to SOC
CNTPSIQ	master	Signal	Timer signals to SOC
cntvalueb	slave	CounterInterface	Interface to SoC level counter module
CNTVIRQ	master	Signal	Timer signals to SOC
commirq	master	Signal	Interrupt signal from debug communication channel.
cp15sdisable	slave	Signal	This signal disables write access to some system control processor registers
cpuporeset	slave	Signal	Power on reset. Initializes all the processor logic, including debug logic.
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti0extin	slave	Signal	-
cti0extout	master	Signal	-
cti1extin	slave	Signal	-
cti1extout	master	Signal	-
cti2extin	slave	Signal	-
cti2extout	master	Signal	-
cti3extin	slave	Signal	-
cti3extout	master	Signal	-
cti	master	v8EmbeddedCrossTrigger_controlprotocol	-
ctidbgirq	master	Signal	-
dbgen	slave	Signal	-
dbgnopwrdown	master	Signal	These signals relate to core power down.
dbgpwrdownack	master	Signal	Debug power down acknowledge.
dbgpwrdownreq	slave	Signal	Debug power down request.
dev_debug_s	slave	PVBus	External debug interface.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
ext_slave_s	slave	PVBus	External Slave port. Equivalent to AXIS port.
external_trace_reset	slave	Signal	ETMv4 External Trace Reset signal.

Port	Direction	Protocol	Description
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.
flash_m	master	PVBus	Flash Port
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.
hiden	slave	Signal	External debug interface.
hniden	slave	Signal	External debug interface.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
irqs	slave	Signal	These signals drive the CPU's interrupt controller interrupt lines.
l2reset	slave	Signal	This signal resets timer and interrupt controller and l2cache
llpp_m	master	PVBus	LLPP (Low-Latency Peripheral Port).
memorymapped_debug_s	slave	PVBus	External debug interface.
niden	slave	Signal	External debug interface.
periphbase	slave	Value_64	This port sets the base address of private peripheral region
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	master	PVBus	The core will generate bus requests on this port.
rei	slave	Signal	Per core RAM Error Interrupt.
reset	slave	Signal	-
romaddr	slave	Value_64	Debug ROM base address.
romaddrv	slave	Signal	Debug ROM base address valid.
rvbar	slave	Value_64	Reset vector base address.
sei	slave	Signal	Per core System Error physical pins
smpnamp	master	Signal	This signals AMP or SMP mode for each core
spiden	slave	Signal	Secure invasive debug enable.
spniden	slave	Signal	Secure non-invasive debug enable.
standbywfe	master	Signal	This signal indicates if a core is in WFE state
standbywfi	master	Signal	This signal indicates if a core is in WFI state
teinit	slave	Signal	This signal provides default exception handling state.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trace_unit_reset	slave	Signal	ETMv4 Trace Unit Reset signal.
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vinithi	slave	Signal	This signal controls of the location of the exception vectors at reset.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Per core virtual System Error physical pins.

## Parameters for AEMv8RMPCT

### **cpuX.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **cpuX.CP15SDISABLE**

Initialize to disable access to some CP15 registers.

Type: `bool`

Default value: `false`

### **cpuX.CP15SDISABLE2**

Initialize to disable access to some CP15 registers (FEAT\_CP15SDISABLE2).

Type: `bool`

Default value: `false`

### **cpuX.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **cpuX.DCZID-log2-block-size**

Log2 of the block size in words cleared by DC ZVA instruction (as read from DCZID\_ELO).

Type: `uint8_t`

Default value: 8

### **cpuX.DCZVA\_single\_write**

Execute the DCZVA as a single write.

Type: `bool`

Default value: `false`

### **cpuX.MPIDR-override**

Override of MPIDR value. If nonzero will override the MT, cluster and CPU ID bits in MPIDR.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.RVBAR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.RVBAR32`**

Reset vector address in AARCH32 when VINITHI is not set and `ignore_rvbar_in_aarch32` is set.

Type: `uint32_t`

Default value: `0x0`

### **`cpuX.SMPnAMP`**

Enable broadcast messages necessary for correct SMP operation at reset.

Type: `bool`

Default value: `true`

### **`cpuX.TEINIT`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **`cpuX.VINITHI`**

Reset value of SCTLR.V.

Type: `bool`

Default value: `false`

### **`cpuX.aarch32_reset_from_impdef_addr`**

If PE resets into AArch32, Whether execution starts from IMPDEF address or hi/low vector.

Type: `bool`

Default value: `true`



**cpuX.ase-present**

Set whether the model has been built with NEON support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**cpuX.clock\_divider**

Clock divider ratio for asymmetric MP clocking.

Type: `uint32_t`

Default value: `0x1`

**cpuX.clock\_multiplier**

Clock divider ratio for asymmetric MP clocking.

Type: `uint32_t`

Default value: `0x1`

**cpuX.crypto\_aes**

AES instructions supported (requires CryptoPlugin to be loaded). 0, not implemented. 2, AES and PMULL instructions implemented (FEAT\_AES, FEAT\_PMULL).

Type: `uint8_t`

Default value: `2`

**cpuX.crypto\_sha1**

SHA-1 instructions supported (requires CryptoPlugin to be loaded). 0, not implemented. 1, SHA1 instructions implemented (FEAT\_SHA1).

Type: `uint8_t`

Default value: `1`

**cpuX.crypto\_sha256**

SHA-256 instructions supported (requires CryptoPlugin to be loaded). 0, not implemented. 1, SHA256 instructions implemented (FEAT\_SHA256).

Type: `uint8_t`

Default value: `1`

**cpuX.cti-intack\_mask**

Set bits represent that the corresponding trigger requires software acknowledge via CTIINTACK.

Type: `uint8_t`

Default value: 1

**cpuX.cti-number\_of\_claim\_bits**

Number of implemented bits in CTICLAIMSET.

Type: `uint8_t`

Default value: 0

**cpuX.cti-number\_of\_triggers**

Number of cti event triggers (default: 8, valid values: {3-32}).

Type: `uint8_t`

Default value: 8

**cpuX.dcache-size**

L1 D-Cache size in bytes.

Type: `uint32_t`

Default value: 0x8000

**cpuX.enable\_crc32**

CRC32 instructions supported. 0, not implemented. 1, CRC32 instructions implemented (FEAT\_CRC32).

Type: `uint8_t`

Default value: 0

**cpuX.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**cpuX.etm-present**

Set whether the model has ETM support.

Type: `bool`

Default value: `true`

**cpuX.flash.enable**

Enable flash by default after reset.

Type: `bool`

Default value: `false`

**cpuX.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `false`

**cpuX.force-fpsid-value**

Value to override the FPSID value to.

Type: `uint32_t`

Default value: `0x0`

**cpuX.has\_hcptr\_tase**

If false, HCPTR.TASE is **RES0**.

Type: `bool`

Default value: `true`

**cpuX.highest-index-of-context-breakpoints**

Highest index of breakpoints that are context aware.

Type: `uint8_t`

Default value: `15`

**cpuX.icache-size**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: `0x8000`

**cpuX.llpp.base**

Sets the base address of Low Latency Peripheral Port.

Type: `uint64_t`

Default value: 0x0

**cpuX.llpp.size**

Sets the size of LLPP(in bytes).

Type: uint32\_t

Default value: 0x1000

**cpuX.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**cpuX.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**cpuX.number-of-breakpoints**

Number of breakpoints.

Type: uint8\_t

Default value: 16

**cpuX.number-of-context-breakpoints**

Number of breakpoints that are context aware.

Type: uint8\_t

Default value: 16

**cpuX.number-of-watchpoints**

Number of watchpoints.

Type: uint8\_t

Default value: 16

**cpuX.operation\_bandwidth**

Operation width for ARMv8.4 PMU extension.

Type: `uint8_t`

Default value: 1

**`cpuX.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**`cpuX.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**`cpuX.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: 0x123456

**`cpuX.semihosting-T32_HLT`**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**`cpuX.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**`cpuX.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**`cpuX.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

### **`cpuX.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`cpuX.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`cpuX.semihosting-prefix`**

Prefix semihosting output with target instance name.

Type: `bool`

Default value: `false`

### **`cpuX.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`cpuX.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`cpuX.semihosting-stderr_istty`**

Result for semihost istty call when argument is stderr.

Type: `bool`

Default value: `true`

### **`cpuX.semihosting-stdin_istty`**

Result for semihost istty call when argument is stdin.

Type: `bool`

Default value: `true`

### **`cpuX.semihosting-stdout_istty`**

Result for semihost istty call when argument is stdout.

Type: `bool`

Default value: `true`

### **`cpuX.semihosting-use_stderr`**

Send stderr from the simulated process to host stderr.

Type: `bool`

Default value: `false`

### **`cpuX.tcm-present`**

Disables the TCMs.

Type: `bool`

Default value: `false`

### **`cpuX.tcm-supports-exclusive`**

Whether TCM supports exclusive access.

Type: `bool`

Default value: `false`

### **`cpuX.tcm.a.base`**

Sets the base address of the ATCM. TCMs are implementation defined in the v8R architecture. TCMs are configurable on the model via parameter only.

Type: `uint64_t`

Default value: `0x0`

**cpuX.tcm.a.enable**

Enable ATCM by default after reset. TCMs are implementation defined in the v8R architecture. TCMs are configurable on the model via parameter only.

Type: `bool`

Default value: `false`

**cpuX.tcm.a.size**

Sets the size of the ATCM(in bytes). The size should be aligned with 4KB. The lower 12 bits will be masked out.

Type: `uint32_t`

Default value: `0x4000`

**cpuX.tcm.a.stretch\_clk**

Whether ATCM clock stretched to occupy full cycle.

Type: `bool`

Default value: `false`

**cpuX.tcm.a.wait**

ATCM accesses wait states.

Type: `uint32_t`

Default value: `0x0`

**cpuX.tcm.b.base**

Sets the base address of the BTCM. TCMs are implementation defined in the v8R architecture. TCMs are configurable on the model via parameter only.

Type: `uint64_t`

Default value: `0x0`

**cpuX.tcm.b.enable**

Enable BTCM by default after reset. TCMs are implementation defined in the v8R architecture. TCMs are configurable on the model via parameter only.

Type: `bool`

Default value: `false`



**cpuX.tcm.b.size**

Sets the size of the BTCM(in bytes). The size should be aligned with 4KB. The lower 12 bits will be masked out.

Type: uint32\_t

Default value: 0x4000

**cpuX.tcm.b.stretch\_clk**

Whether BTCM clock stretched to occupy full cycle.

Type: bool

Default value: false

**cpuX.tcm.b.wait**

BTCM accesses wait states.

Type: uint32\_t

Default value: 0x0

**cpuX.tcm.c.base**

Sets the base address of the CTCM. TCMs are implementation defined in the v8R architecture. TCMs are configurable on the model via parameter only.

Type: uint32\_t

Default value: 0x0

**cpuX.tcm.c.enable**

Enable CTCM by default after reset. TCMs are implementation defined in the v8R architecture. TCMs are configurable on the model via parameter only.

Type: bool

Default value: false

**cpuX.tcm.c.size**

Sets the size of the CTCM(in bytes). The size should be aligned with 4KB. The lower 12 bits will be masked out.

Type: uint32\_t

Default value: 0x2000

**cpuX.tcm.c.stretch\_clk**

Whether CTCM clock stretched to occupy full cycle.

Type: `bool`

Default value: `false`

**cpuX.tcm.c.wait**

CTCM accesses wait states.

Type: `uint32_t`

Default value: `0x0`

**cpuX.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**cpuX.unpredictable\_WPMASKANDBAS**

Constrained unpredictable handling of watchpoints when mask and BAS fields specified. 0, IGNOREMASK. 1, IGNOREBAS (default). 2, REPEATBAS8. 3, REPEATBAS.

Type: `uint8_t`

Default value: `1`

**cpuX.vfp-dp-present**

Whether double-precision floating point feature is implemented (FEAT\_F64MM).

Type: `bool`

Default value: `true`

**cpuX.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**cpuX.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**cpuX.vfp-traps**

Implement support for trapping floating-point exceptions.

Type: `bool`

Default value: `true`

**cpuX.vfp-traps-show-all**

Report all trapped floating-point exceptions in the syndrome when a combination occurs.

Type: `bool`

Default value: `false`

**cpuX.wfet\_early\_or\_delayed\_timeout**

WFET early or delayed timeout beyond the threshold value of CNTVCT\_ELO in percentage.

Type: `int8_t`

Default value: `0`

**cpuX.wfit\_early\_or\_delayed\_timeout**

WFIT early or delayed timeout beyond the threshold value of CNTVCT\_ELO in percentage.

Type: `int8_t`

Default value: `0`

**ADFSR-AIFSR-implemented**

ADFSR and AIFSR are implemented.

Type: `bool`

Default value: `false`

**AIDR**

Value of AIDR\_EL1 register.

Type: `uint64_t`

Default value: 0x0

**AMIIDR**

Value of AMU Implementation Identification Register.

Type: uint64\_t

Default value: 0x43b

**AMPIDR**

Value of AMU Peripheral Identification Register.

Type: uint64\_t

Default value: 0x4000bb000

**BPIMVA\_causes\_translation\_lookup**

Do a translation when BPIMVA instruction is executed (which may cause a translation fault).

Type: bool

Default value: false

**BROADCASTATOMIC**

Enable broadcasting of atomic operation to NC/DEV memory. The broadcastatomic signal will override this value if used.

Type: bool

Default value: true

**BROADCASTATOMICL**

Enable broadcasting of atomic operation to LLRAM memory. The broadcastatomicl signal will override this value if used.

Type: bool

Default value: true

**BROADCASTCACHEMINT**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

Type: bool

Default value: true

**BROADCASTINNER**

Enable broadcasting of Inner Shareable transactions. The broadcastinner signal will override this value if used.

Type: `bool`

Default value: `true`

**BROADCASTOUTER**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

Type: `bool`

Default value: `true`

**CCSIDR-L1D\_override**

If nonzero, override the value presented in CCSIDR for L1D (this is cosmetic and does not affect cache behaviour).

Type: `uint64_t`

Default value: `0x0`

**CCSIDR-L1I\_override**

If nonzero, override the value presented in CCSIDR for L1I (this is cosmetic and does not affect cache behaviour).

Type: `uint64_t`

Default value: `0x0`

**CCSIDR-L2\_override**

If nonzero, override the value presented in CCSIDR for L2 (this is cosmetic and does not affect cache behaviour).

Type: `uint64_t`

Default value: `0x0`

**CCSIDR-L3\_override**

If nonzero, allow L3 selection in CSSELR and present this value in CCSIDR (this is cosmetic and does not affect cache behaviour).

Type: `uint64_t`

Default value: `0x0`

**CFGTFPEN\_pin\_reset**

CFGTFPEN Configuration pin at reset for bitfield IMP\_MEMPROTCTLR\_EL1.TFPEN.

Type: `bool`

Default value: `false`

**CHI**

Selects the type of protocol the Main Manager(MM) interface implements. 0, MM port configured as AXI. 1, MM port configured as CHI.

Type: `bool`

Default value: `false`

**CLUSTER\_ID**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the ManagerID64 of the core.

Type: `uint32_t`

Default value: `0x0`

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. This value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain even when cache state modelling is disabled. Possible values are:- 0 = All CMOs are broadcast if architecturally required- 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required.

Type: `uint8_t`

Default value: `1`

**CPUCFR**

Value of CPU Configuration Register.

Type: `uint64_t`

Default value: `0x0`

**CTIPIDR**

If non-zero, override the CTI Peripheral Identification Register.

Type: `uint64_t`

Default value: `0x0`

**CTR-L1Ip-override**

If non-zero, override the L1Ip bits in CTR/CTR\_ELO system register. This does not change the behaviour of the cache, only what is present in the CTR register.

Type: `uint8_t`

Default value: `0`

**DBGBCR\_BT\_applies\_RES0\_before\_valid\_check**

If true, **RES0** behaviour is applied to DBGBCR(\_EL1).BT before checking for reserved values for this field.

Type: `bool`

Default value: `true`

**DBGPIDR**

If non-zero, override the Debug Peripheral Identification Register.

Type: `uint64_t`

Default value: `0x0`

**DBGROMADDR**

Initialization value of DBGDRAR register. Bits[55:12] of this register specify the ROM table physical address.

Type: `uint64_t`

Default value: `0x0`

**DBGROMADDRV**

If true, set bits[1:0] of the CP15 DBGDRAR to indicate that the address is valid.

Type: `bool`

Default value: `false`

**ERRIIDR**

Value of RAS Implementation Identification Register.

Type: `uint64_t`

Default value: `0xd800143b`

### **ERRPIDR**

Value of RAS Peripheral Identification Register.

Type: `uint64_t`

Default value: `0x4100bbd80`

### **ERXMISC0\_mask**

Write Mask for ERXMISC0 RAS Register.

Type: `uint64_t`

Default value: `0x0`

### **GICDISABLE**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`

Default value: `true`

### **MIDR**

Value of MIDR\_EL1 register.

Type: `uint32_t`

Default value: `0x410fd0f0`

### **NUM\_CORES**

Number of cores in cluster.

Type: `uint8_t`

Default value: `1`

### **PA\_SIZE**

Physical address range supported (FEAT\_LPA).

Type: `uint8_t`

Default value: `40`



**PERIPHBASE**

Base address of peripheral memory space.

Type: `uint64_t`

Default value: `0x13080000`

**PMCEID0**

Performance Monitor Common Event ID Reg 0 value - 64 bit.

Type: `uint64_t`

Default value: `0xffffffff`

**PMCEID1**

Performance Monitor Common Event ID Reg 1 value - 64 bit.

Type: `uint64_t`

Default value: `0xffffffff`

**PMSIDR.ArchInst**

Defines whether architecture instruction sampling is implemented or not, if not only micro op sampling is implemented. Model only supports architecture instruction sampling, but allows ID register field to be configured.

Type: `bool`

Default value: `true`

**PMSIDR.CRR**

Defines whether call return branch records (FEAT\_SPE\_CRR) is implemented or not.

Type: `bool`

Default value: `false`

**PMSIDR.LDS**

Defines whether data source for sampled load instruction is implemented or not. Model does not implement loaded data source, but allows ID register field to be configured.

Type: `bool`

Default value: `false`

**PMUPIDR**

If non-zero, override the PMU Peripheral Identification Register.

Type: `uint64_t`

Default value: `0x0`

### **VMSA\_supported**

VMSA is supported at EL1.

Type: `bool`

Default value: `true`

### **abort\_execution\_from\_device\_memory**

Execution from device memory generates a prefetch abort.

Type: `bool`

Default value: `false`

### **advsimd\_overread**

AdvSIMD element load operations access all bytes of a 16-byte aligned window, even in Device memory.

Type: `bool`

Default value: `false`

### **align\_pc\_on\_branch\_to\_unaligned\_pc\_aarch32**

Force PC align for branches to an unaligned PC counter in A32 state.

Type: `bool`

Default value: `false`

### **align\_pc\_on\_debug\_exit\_to\_aarch32**

Exit to AARCH32 state from debug state forces pc bit0 to 0.

Type: `bool`

Default value: `false`

### **align\_pc\_on\_illegal\_exception\_return\_to\_aarch32**

Align PC when performing an illegal exception return from AArch64 to AArch32.

Type: `bool`

Default value: `true`

**amu\_aux\_type\_fixed**

Lists which AMU auxiliary registers that are fixed and to which event type. The JSON schema is: {fixed\_aux\_reg:evt\_type, ...}. For example {"0":0x300} would make auxiliary register 0 fixed to event type 0x300.

Type: string

Default value: N/A

**amu\_mmap\_address**

AMU base address for each core on system bus. 0 means the AMU is not mapped, otherwise the address must be 4KB aligned. JSON schema for the parameter value is: {"format":"all\_addrs\_are\_absolute\_wrt\_systembus", "cores": [{"amu":0x0}, {"amu":0x0}, {"amu":0x0}, {"amu":0x0}]}.

Type: string

Default value: N/A

**amu\_num\_auxiliary\_counters**

Number of AMU auxiliary counters implemented.

Type: uint8\_t

Default value: 0

**apsr\_read\_restrict**

At ELO, unknown bits of APSR are **RAZ**.

Type: bool

Default value: false

**atomic\_memtype\_fault\_prio\_less\_than\_gpc\_fault**

Bitmask controlling whether unsupported memtype faults are lower priority than GPC faults: bit0 applies to atomic/exclusive accesses, bit1 applies to LS64.

Type: uint8\_t

Default value: 0

**atomic\_memtype\_fault\_priority**

This parameter describes the priority of unsupported atomic/exclusive memtype fault w.r.t alignment and permission fault. 0, BEFORE\_ALIGN\_MEM\_FAULT. 1, AFTER\_ALIGN\_BEFORE\_PERM\_FAULT. 2, AFTER\_PERM\_FAULT.

Type: uint8\_t

Default value: 0

### **auxilliary\_feature\_register0**

Value of AFR0 ID register.

Type: `uint32_t`

Default value: `0x0`

### **branch-predictor-clear-policy**

Set branch prediction policy as defined for MMFR1[31:28]. This does not change the behaviour of the branch predictor, only what is reported in MMFR1.BPred.

Type: `uint8_t`

Default value: 2

### **branch-predictor-supported-ops**

Set branch prediction policy as defined for MMFR3[11:8]. This does not change the behaviour of the branch predictor, only what is reported in MMFR3.BPMaint.

Type: `uint8_t`

Default value: 1

### **bus\_protection\_enable\_at\_reset**

Enable TCM, L1Cache bus protection after reset.

Type: `bool`

Default value: `false`

### **cache-log2linelen**

Log2 of the cache line length in bytes.

Type: `uint8_t`

Default value: 6

### **cache\_maintenance\_hits\_watchpoints**

DCIMVA operations executed in AArch32 modes hit watchpoints.

Type: `bool`

Default value: `false`

**changing\_block\_size\_without\_bbm\_support**

Level of support for changing block size without break-before-make (FEAT\_BBM).

Type: `uint8_t`

Default value: 0

**check\_memory\_attributes**

Detect and report TLB use of conflicting memory attributes for views of the same physical address.

Type: `bool`

Default value: `false`

**clean\_invalidate\_cache\_on\_warm\_reset**

Clean and invalidate caches on warm reset.

Type: `bool`

Default value: `false`

**clear\_reg\_top\_eret**

Behaviour of the upper 32-bits of the Xn registers when changing between AArch32 state and AArch64 state. 0, upper 32-bits preserved for all registers. 1, upper 32-bits set to 0 for all accessible registers. 2, upper 32-bits set to 0 for a random selection of accessible registers. 3, upper-32-bits set to 0 for registers touched in AArch32.

Type: `uint8_t`

Default value: 1

**clear\_reg\_top\_set**

Whether to clear upper 32-bits of the Xn register when corresponding AArch32 register is set via Iris.

Type: `bool`

Default value: `true`

**cluster\_utid**

Unique cluster transaction identifier for interconnect protection.

Type: `uint8_t`

Default value: 0

**configure\_pmu\_events\_with\_json**

"Configure v8.6 and newer PMU events. Note : This param has high priority and overrides the setting of "has\_\*\_pmu\_events" if both the params are provided. JSON schema for the parameter value is e.g. 1. {"all":false} 2. {"pmu\_events":["EVENT\_NAME\_1","EVENT\_NAME\_2"]}".

Type: string

Default value: N/A

**configure\_v8\_6\_pmu\_events\_with\_json**

"[DEPRECATED: Set configure\_pmu\_events\_with\_json to the same value instead] Configure v8.6 PMU events. Note : This param has high priority and overrides the setting of "has\_v8\_6\_pmu\_events" if both the params are provided. JSON schema for the parameter value is e.g. 1. {"all":false} 2. {"pmu\_events":["BR\_INDNR\_RETIRE", "BR\_IND\_RETIRE", "BR\_RETURN\_SKIP\_RETIRE", "BR\_RETURN\_ANY\_RETIRE", "BR\_INDNR\_SKIP\_RETIRE", "BR\_INDNR\_TAKEN\_RETIRE", "BR\_IND\_SKIP\_RETIRE", "BR\_IND\_TAKEN\_RETIRE", "BR\_IMMED\_SKIP\_RETIRE", "BR\_IMMED\_TAKEN\_RETIRE", "BR\_SKIP\_RETIRE"]}".

Type: string

Default value: N/A

**configure\_v8\_8\_pmu\_events\_with\_json**

"[DEPRECATED: Set configure\_pmu\_events\_with\_json to the same value instead] Configure v8.8 PMU events. Note : This param has high priority and overrides the setting of "has\_v8\_8\_pmu\_events" if both the params are provided. JSON schema for the parameter value is e.g. 1. {"all":false} 2. {"pmu\_events":["BR\_HINT\_COND\_RETIRE", "BR\_COND\_TAKEN\_RETIRE", "BR\_UNCOND\_RETIRE", "BR\_COND\_RETIRE", "BRNL\_TAKEN\_RETIRE", "BRNL\_IND\_TAKEN\_RETIRE", "BRNL\_INDNR\_TAKEN\_RETIRE", "BRNL\_IMMED\_TAKEN\_RETIRE", "BL\_TAKEN\_RETIRE", "BL\_IND\_TAKEN\_RETIRE", "BL\_IMMED\_TAKEN\_RETIRE"]}".

Type: string

Default value: N/A

**configure\_v8\_9\_pmu\_events\_with\_json**

"[DEPRECATED: Set configure\_pmu\_events\_with\_json to the same value instead] Configure v8.9 PMU events. Note : This param has high priority and overrides the setting of "has\_v8\_9\_pmu\_events" if both the params are provided. JSON schema for the parameter value is e.g. 1. {"all":false} 2. {"pmu\_events":["ASE\_SVE\_RETIRE", "ASE\_RETIRE", "VFP\_RETIRE", "SVE\_RETIRE", "CRYPTO\_RETIRE", "SIMD\_INST\_RETIRE", "ASE\_INST\_RETIRE", "SVE\_INST\_RETIRE", "ASE\_SVE\_INST\_RETIRE", "LD\_ANY\_RETIRE", "ST\_ANY\_RETIRE", "LDST\_ANY\_RETIRE", "DP\_RETIRE"]}".

Type: string

Default value: N/A

**core\_cache\_protection**

core\_cache\_protection can change ERROFR, ERROPFGF and ERROPFGCTL fields. Possible values are: -1:Not implemented (by default), 0:Disabled, 1:Enabled.

Type: int8\_t

Default value: -1

**cpacr\_trcdis\_behaviour**

Behaviour of CPACR.TRCDIS/NSACR.NSTRCDIS when there is no CP14 ETM interface. 0, **RAZ/WI**. 2, implemented.

Type: uint8\_t

Default value: 2

**cpu\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: uint32\_t

Default value: 0x1

**cpu\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: uint32\_t

Default value: 0x1

**dbg-bcr-reserved-behavior**

This is the behavior of the reserved values of the BT field in DBGBCRPossible values are: - 0 = Disabled.- 1 = BT[2] is ignored..

Type: uint8\_t

Default value: 1

**dbg\_rom\_dap\_addr**

Debug ROM dap base address.

Type: uint64\_t

Default value: 0x0

**dbgitr\_buffer\_size**

Number of instructions which can be buffered before EDSCR.ITE is cleared.

Type: `uint32_t`

Default value: `0x0`

### **`dbgxvr_ress_is_stateful`**

Whether DBGWVR/DBGBVR.RESS returns last written value. if set to false, RESS returns sign extended value.

Type: `bool`

Default value: `false`

### **`dc_fault_unaligned_s1_device_s2_fwb`**

Whether takes an Alignment Fault caused by the memory type on a DC {ZVA,GZVA,GVA} if the stage 1 memory type is any Device memory type.

Type: `bool`

Default value: `false`

### **`dcache-hit_latency`**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`dcache-maintenance_latency`**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`dcache-miss_latency`**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`dcache-prefetch_enabled`**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

Type: `bool`



Default value: `false`

### **`dcache-read_access_latency`**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`dcache-read_bus_width_in_bytes`**

L1 D-Cache read bus width in bytes used to calculate per-access timing annotations.

Type: `uint32_t`

Default value: `0x8`

### **`dcache-read_latency`**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`dcache-size`**

L1 D-Cache size in bytes.

Type: `uint32_t`

Default value: `0x8000`

### **`dcache-snoop_data_transfer_latency`**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`dcache-state_modelled`**

Set whether D-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **`dcache-ways`**

L1 D-Cache number of ways (sets are implicit from size).

Type: `uint8_t`

Default value: 2

### **`dcache-write_access_latency`**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`dcache-write_bus_width_in_bytes`**

L1 D-Cache write bus width in bytes used to calculate per-access timing annotations.

Type: `uint32_t`

Default value: `0x8`

### **`dcache-write_latency`**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`dcimva_requires_s2_write_permissions`**

Data-cache invalidate by MVA operations require stage 2 write permission (virtualised AArch32 guest).

Type: `bool`

Default value: `false`

### **`debug_auth_signals_sampled_at_reset`**

Debug authentication signals can be configured as either sampled at reset only or at any time for External Root Debug.

Type: `bool`

Default value: `false`

### **debug\_components\_dap\_address**

Debug components ROM,ED,CTI,PMU,TRACE and TRBU base address for each core on debug bus. The "rom" field in the "cores" array are only allowed when 'debug\_rom\_is\_flat' is false. JSON schema for the parameter value is: {"format":"all\_addrs\_are\_absolute\_wrt\_debugbus", "cores": [{"rom":0x0, "ed":0x0, "cti":0x0, "pmu":0x0, "etm":0x0, "trbu":0x0}, {"rom":0x0, "ed":0x0, "cti":0x0, "pmu":0x0, "etm":0x0, "trbu":0x0}, {"rom":0x0, "ed":0x0, "cti":0x0, "pmu":0x0, "etm":0x0, "trbu":0x0}, {"rom":0x0, "ed":0x0, "cti":0x0, "pmu":0x0, "etm":0x0, "trbu":0x0}]}

Type: `string`

Default value: `N/A`

### **debug\_components\_mmap\_address**

Debug components ROM,ED,CTI,PMU,TRACE and TRBU base address for each core on system bus. The "rom" field in the "cores" array are only allowed when 'debug\_rom\_is\_flat' is false. JSON schema for the parameter value is: {"format":"all\_addrs\_are\_absolute\_wrt\_systembus", "cores": [{"rom":0x0, "ed":0x0, "cti":0x0, "pmu":0x0, "etm":0x0, "trbu":0x0}, {"rom":0x0, "ed":0x0, "cti":0x0, "pmu":0x0, "etm":0x0, "trbu":0x0}, {"rom":0x0, "ed":0x0, "cti":0x0, "pmu":0x0, "etm":0x0, "trbu":0x0}, {"rom":0x0, "ed":0x0, "cti":0x0, "pmu":0x0, "etm":0x0, "trbu":0x0}]}

Type: `string`

Default value: `N/A`

### **debug\_entry\_is\_context\_sync**

If true, Entry in debug state is Context sync. Exiting debug state is a context synchronising operation, but entering is not. However some cpu implementation can consider also the Entry in Debug state as a CSE.

Type: `bool`

Default value: `false`

### **debug\_rom\_is\_class\_9**

If true, present a debug ROM table as a class 9 device. Otherwise, use a class 1 ROM table.

Type: `bool`

Default value: `false`

### **debug\_rom\_is\_flat**

If true, present a debug ROM table recommended by ARMv8 Debug Architecture. Otherwise, use nested ROM tables.

Type: `bool`

Default value: `false`

### **def\_mem\_map**

Default memory map in a json format which is: Where the \* represents the entire physical address range and must be provided and the attributes can be a combination of following. MemoryType - NORMAL, GRE, nGRE, nGnRE, nGnRnEShareability - ISH, OSH, NSHInnerAttributes - IWB, IWT, INCOuterAttribute - OWB, OWT, ONCEExecuteNever - XN.

Type: `string`

Default value: `"{\": \"NORMAL INC ONC OSH\"}"`

### **def\_mem\_map\_file\_path**

Path of file describing default memory map in json format. When a valid path is provided, the below parameter 'def\_mem\_map' will be ignored.

Type: `string`

Default value: `N/A`

### **default\_inner\_shareable**

shareability for default memory map regions which are shareable.

Type: `bool`

Default value: `false`

### **delay\_serror**

Add a propagation delay of serror signal into the core.

Type: `uint32_t`

Default value: `0x0`

### **dic-spi\_count**

Number of shared peripheral interrupts implemented.

Type: `uint8_t`

Default value: `64`

### **disable\_impdef\_abort\_on\_ic\_maintenance**

Disable IMP\_INTLATENCY\_EL2.MMDVM/LLRAMDVM controlling the abort on IC maintenance on MM port/LLRAM.

Type: `bool`

Default value: `false`

### **`disable_sve_plugin`**

If true, SVE will not be implemented in this processor even if the plugin is loaded (FEAT\_SVE).

Type: `bool`

Default value: `false`

### **`disable_unknown_update_event_on_reset`**

Disables SYSREG\_UPDATE event notification on reset for the registers whose bitfields are all reserved or resets to architecturally unknown value.

Type: `bool`

Default value: `false`

### **`dsb_accumulate_threshold`**

Limit the maximum number of observable DSB side effects which can be queued (e.g. TLBI), after which DSB sync will be done automatically.

Type: `uint32_t`

Default value: `0x100`

### **`edpfr_ras_unknown_bits_read_as_0`**

If true then **UNKNOWN** bits in RAS field in EDPFR are read as 0.

Type: `bool`

Default value: `false`

### **`el0_can_access_imp_def_functionality`**

If not made UNDEF by `imp_def_functionality_behaviour`, EL0 can access **IMPLEMENTATION DEFINED** registers and system instructions.

Type: `bool`

Default value: `false`

### **`el3_trap_priority_when_secure_debug_disabled`**

Undef when secure debug is disabled (EDSCR.SDD == 1) && boolean **IMPLEMENTATION\_DEFINED** 'EL3 trap priority when SDD == 1'.

Type: `bool`

Default value: `false`

**enable-gicv5**

if enable-gicv5 is set, then GICv5 is Supported.

Type: `bool`

Default value: `false`

**enable\_address\_contig\_check**

Check the input address range for the table entries that have the contiguous hint bit set.

Type: `bool`

Default value: `false`

**enable\_debug\_auth\_signals\_config**

Debug Authentication Signals DBGEN, SPIDEN (and if RME is enabled RLPIDEN and RTPIDEN) are configurable (default) or not configurable, (hardwired to 1). This parameter is the integer representation of a bitmap to enable configuration of these signals, with:- BIT[0] = DBGEN- BIT[1] = SPIDEN- BIT[2] = RLPIDEN- BIT[3] = RTPIDEN.

Type: `uint8_t`

Default value: 15

**enable\_lock\_step**

Whether the core is configured in Dual Core Lock Step mode (FEAT\_DCLS).

Type: `bool`

Default value: `false`

**enable\_tlb\_contig\_check**

Perform extra pagetable walks to check translation table entries that have the contiguous hint bit set. Ignored if FEAT\_BBML3 is implemented.

Type: `bool`

Default value: `false`

**enhanced\_pac2\_level**

Implements Enhanced PAC2. 0: No EnhancedPAC2, 1: EnhancedPAC2 Only, 2: EnhancedPAC2 with FPAC, 3: EnhancedPAC2 with FPACCombined.

Type: `uint8_t`

Default value: 0

**error\_record\_feature\_register**

RAS feature register values. An array of JSON objects. The JSON schema for the array is: [{"ED":0x0, "IMPDEF\_3\_2":0x0, "UI":0x0, "FI":0x0, "UE":0x0, "CFI":0x0, "CEC":0x0, "RP":0x0, "DUI":0x0, "CEO":0x0, "CI":0x0, "TS":0x0, "INJ":0x0, "FRX":0x0, "UC":0x0, "UEU":0x0, "UER":0x0, "UEO":0x0, "DE":0x0, "CE":0x0, "Visibility":"Core"},other\_feature\_register\_values]. Where ED,UI,FI,CE and UE have valid values between 0x0 - 0x3. CFI and DUI have valid values 0x0, 0x2 and 0x3. CEC has valid values 0x0,0x2 or 0x4. RP,CEO,INJ,FRX,UC,UEU,UER,UEO,DE has valid values 0x0 or 0x1. CI and TS has valid values of 0x0, 0x1 and 0x2. Visibility has valid values "Core" or "Cluster".

Type: string

Default value: N/A

**error\_record\_feature\_register\_json\_file**

File path to the RAS feature register values as JSON. The file uses the same format as the error\_record\_feature\_register parameter value.

Type: string

Default value: N/A

**erxpfctl\_res0\_stateful\_mask**

Mask for stateful bits for ERXPFGCTL which are **RES0**.

Type: uint64\_t

Default value: 0x0

**exception\_catch\_before\_software\_step**

Exception catch priority for the exception trapping form of exception catch (Armv8.2 or later, or exception\_catch\_type=0). If true, the exception catch debug event has higher priority than software step and halting step.

Type: bool

Default value: true

**exception\_catch\_type**

Type of exception catch (ARMv8.0 - ARMv8.1 only). 0, exception trapping. 1, non-exception trapping, higher priority than step. 2, non-exception trapping, lower priority than step.

Type: uint8\_t

Default value: 0

**exclusive\_monitor\_clear\_on\_atomic\_from\_same\_master**

Exclusive monitors in the cluster will be cleared by a atomic by the same master to the monitored address.

Type: `bool`

Default value: `true`

**exclusive\_monitor\_clear\_on\_store\_from\_same\_master**

Exclusive monitors in the cluster will be cleared by a store by the same master to the monitored address.

Type: `bool`

Default value: `true`

**exclusive\_monitor\_clear\_on\_strex\_address\_mismatch**

Exclusive monitors in the cluster will be cleared when a strex fails because the address does not match.

Type: `bool`

Default value: `true`

**exclusive\_monitor\_clear\_on\_strex\_success**

Exclusive monitors in the cluster will be cleared when a strex succeeds.

Type: `bool`

Default value: `true`

**exercise\_stxr\_fail**

Controls the rejection of exclusive store instructions. 0: exclusive store instructions should behave as normal, 1: Reject a pseudo-random majority of exclusive store instructions, 2: Always fail exclusive store instructions.

Type: `uint8_t`

Default value: 0

**ext\_abort\_device\_GRE\_prefetch\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as `ext_abort_prefetch_ras_index`, Valid indices in range [0, `number_of_error_records`-1].

Type: `int16_t`

Default value: -1



**ext\_abort\_device\_GRE\_prefetch\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as ext\_abort\_prefetch\_ras\_type, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO , 4 = UER, 5 = CE.

Type: `int8_t`

Default value: -1

**ext\_abort\_device\_GRE\_read\_is\_critical**

Critical reporting of device-GRE read external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_device\_GRE\_read\_is\_sync**

Synchronous reporting of device-GRE read external aborts. 0, asynchronous. 1, synchronous. 2, same as ext\_abort\_device\_read\_is\_sync.

Type: `uint8_t`

Default value: 2

**ext\_abort\_device\_GRE\_read\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as ext\_abort\_device\_read\_ras\_index, Valid indices in range [0, number\_of\_error\_records-1].

Type: `int16_t`

Default value: -1

**ext\_abort\_device\_GRE\_read\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as ext\_abort\_device\_read\_ras\_type, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO , 4 = UER, 5 = CE.

Type: `int8_t`

Default value: -1

**ext\_abort\_device\_GRE\_write\_is\_critical**

Critical reporting of device-GRE write external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_device\_GRE\_write\_is\_sync**

Synchronous reporting of device-GRE write external aborts. 0, asynchronous. 1, synchronous. 2, same as ext\_abort\_device\_write\_is\_sync.

Type: uint8\_t

Default value: 2

**ext\_abort\_device\_GRE\_write\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as ext\_abort\_device\_write\_ras\_index, Valid indices in range [0, number\_of\_error\_records-1].

Type: int16\_t

Default value: -1

**ext\_abort\_device\_GRE\_write\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as ext\_abort\_device\_write\_ras\_type, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: int8\_t

Default value: -1

**ext\_abort\_device\_nGRE\_prefetch\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as ext\_abort\_prefetch\_ras\_index, Valid indices in range [0, number\_of\_error\_records-1].

Type: int16\_t

Default value: -1

**ext\_abort\_device\_nGRE\_prefetch\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as ext\_abort\_prefetch\_ras\_type, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: int8\_t

Default value: -1

**ext\_abort\_device\_nGRE\_read\_is\_critical**

Critical reporting of device-nGRE read external aborts.

Type: bool

Default value: false

**ext\_abort\_device\_nGRE\_read\_is\_sync**

Synchronous reporting of device-nGRE read external aborts. 0, asynchronous. 1, synchronous. 2, same as ext\_abort\_device\_read\_is\_sync.

Type: uint8\_t

Default value: 2

**ext\_abort\_device\_nGRE\_read\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as ext\_abort\_device\_read\_ras\_index, Valid indices in range [0, number\_of\_error\_records-1].

Type: int16\_t

Default value: -1

**ext\_abort\_device\_nGRE\_read\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as ext\_abort\_device\_read\_ras\_type, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: int8\_t

Default value: -1

**ext\_abort\_device\_nGRE\_write\_is\_critical**

Critical reporting of device-nGRE write external aborts.

Type: bool

Default value: false

**ext\_abort\_device\_nGRE\_write\_is\_sync**

Synchronous reporting of device-nGRE write external aborts. 0, asynchronous. 1, synchronous. 2, same as ext\_abort\_device\_write\_is\_sync.

Type: uint8\_t

Default value: 2

**ext\_abort\_device\_nGRE\_write\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as ext\_abort\_device\_write\_ras\_index, Valid indices in range [0, number\_of\_error\_records-1].

Type: int16\_t

Default value: -1

**ext\_abort\_device\_nGRE\_write\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as ext\_abort\_device\_write\_ras\_type, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO , 4 = UER, 5 = CE.

Type: `int8_t`

Default value: -1

**ext\_abort\_device\_prefetch\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as ext\_abort\_prefetch\_ras\_index, Valid indices in range [0, number\_of\_error\_records-1].

Type: `int16_t`

Default value: -1

**ext\_abort\_device\_prefetch\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as ext\_abort\_prefetch\_ras\_type, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO , 4 = UER, 5 = CE.

Type: `int8_t`

Default value: -1

**ext\_abort\_device\_read\_acquire\_is\_sync**

Synchronous reporting of device read with acquire external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_device\_read\_is\_critical**

Critical reporting of device-nGnRE read external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_device\_read\_is\_sync**

Synchronous reporting of device-nGnRE read external aborts.

Type: `bool`

Default value: `true`

**ext\_abort\_device\_read\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number\_of\_error\_records-1].

Type: `uint16_t`

Default value: `0x0`

**ext\_abort\_device\_read\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `uint8_t`

Default value: `0`

**ext\_abort\_device\_write\_is\_critical**

Critical reporting of device-nGnRE write external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_device\_write\_is\_sync**

Synchronous reporting of device-nGnRE write external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_device\_write\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number\_of\_error\_records-1].

Type: `uint16_t`

Default value: `0x0`

**ext\_abort\_device\_write\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `uint8_t`

Default value: `0`

**ext\_abort\_fill\_data**

Returned data, if external aborts are asynchronous.

Type: `uint64_t`

Default value: `0xfdfdfdfdfdfdfdfdf`

**ext\_abort\_in\_virtual\_tag\_reads\_is\_sync**

Behaviour of external aborts generated by virtual tag reads. 0, asynchronous abort. 1, synchronous abort. 2, same as `ext_abort_normal_cacheable_read_is_sync`.

Type: `uint8_t`

Default value: 2

**ext\_abort\_in\_virtual\_tag\_writes\_is\_sync**

Behaviour of external aborts generated by virtual tag writes. 0, asynchronous abort. 1, synchronous abort. 2, same as `ext_abort_normal_cacheable_write_is_sync`.

Type: `uint8_t`

Default value: 2

**ext\_abort\_normal\_cacheable\_read\_is\_critical**

Critical reporting of normal write-back cacheable-read external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_normal\_cacheable\_read\_is\_sync**

Synchronous reporting of normal write-back cacheable-read external aborts.

Type: `bool`

Default value: `true`

**ext\_abort\_normal\_cacheable\_read\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range `[0, number_of_error_records-1]`.

Type: `uint16_t`

Default value: `0x0`

**ext\_abort\_normal\_cacheable\_read\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO , 4 = UER, 5 = CE.

Type: `uint8_t`

Default value: 0

**ext\_abort\_normal\_cacheable\_write\_is\_critical**

Critical reporting of normal write-back cacheable write external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_normal\_cacheable\_write\_is\_sync**

Synchronous reporting of normal write-back cacheable write external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_normal\_cacheable\_write\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number\_of\_error\_records-1].

Type: `uint16_t`

Default value: 0x0

**ext\_abort\_normal\_cacheable\_write\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO , 4 = UER, 5 = CE.

Type: `uint8_t`

Default value: 0

**ext\_abort\_normal\_noncacheable\_prefetch\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as `ext_abort_prefetch_ras_index`, Valid indices in range [0, number\_of\_error\_records-1].

Type: `int16_t`

Default value: -1

**ext\_abort\_normal\_noncacheable\_prefetch\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as ext\_abort\_prefetch\_ras\_type, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `int8_t`

Default value: `-1`

**ext\_abort\_normal\_noncacheable\_read\_is\_critical**

Critical reporting of normal noncacheable-read external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_normal\_noncacheable\_read\_is\_sync**

Synchronous reporting of normal noncacheable-read external aborts.

Type: `bool`

Default value: `true`

**ext\_abort\_normal\_noncacheable\_read\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number\_of\_error\_records-1].

Type: `uint16_t`

Default value: `0x0`

**ext\_abort\_normal\_noncacheable\_read\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `uint8_t`

Default value: `0`

**ext\_abort\_normal\_noncacheable\_write\_is\_critical**

Critical reporting of normal noncacheable write external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_normal\_noncacheable\_write\_is\_sync**

Synchronous reporting of normal noncacheable write external aborts.



Type: `bool`

Default value: `false`

#### **`ext_abort_normal_noncacheable_write_ras_index`**

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number\_of\_error\_records-1].

Type: `uint16_t`

Default value: `0x0`

#### **`ext_abort_normal_noncacheable_write_ras_type`**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO , 4 = UER, 5 = CE.

Type: `uint8_t`

Default value: `0`

#### **`ext_abort_normal_wt_cacheable_prefetch_ras_index`**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as `ext_abort_prefetch_ras_index`, Valid indices in range [0, number\_of\_error\_records-1].

Type: `int16_t`

Default value: `-1`

#### **`ext_abort_normal_wt_cacheable_prefetch_ras_type`**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as `ext_abort_prefetch_ras_type`, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO , 4 = UER, 5 = CE.

Type: `int8_t`

Default value: `-1`

#### **`ext_abort_normal_wt_cacheable_read_is_critical`**

Critical reporting of normal write-through cacheable-read external aborts.

Type: `bool`

Default value: `false`

#### **`ext_abort_normal_wt_cacheable_read_is_sync`**

Synchronous reporting of normal write-through read external aborts. 0, asynchronous. 1, synchronous. 2, same as `ext_abort_normal_cacheable_read_is_sync`.

Type: `uint8_t`

Default value: 2

#### **ext\_abort\_normal\_wt\_cacheable\_read\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as ext\_abort\_normal\_cacheable\_read\_ras\_index, Valid indices in range [0, number\_of\_error\_records-1].

Type: int16\_t

Default value: -1

#### **ext\_abort\_normal\_wt\_cacheable\_read\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as ext\_abort\_normal\_cacheable\_read\_ras\_type, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: int8\_t

Default value: -1

#### **ext\_abort\_normal\_wt\_cacheable\_write\_is\_critical**

Critical reporting of normal write-through write external aborts.

Type: bool

Default value: false

#### **ext\_abort\_normal\_wt\_cacheable\_write\_is\_sync**

Synchronous reporting of normal write-through write external aborts. 0, asynchronous. 1, synchronous. 2, same as ext\_abort\_normal\_cacheable\_write\_is\_sync.

Type: uint8\_t

Default value: 2

#### **ext\_abort\_normal\_wt\_cacheable\_write\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as ext\_abort\_normal\_cacheable\_write\_ras\_index, Valid indices in range [0, number\_of\_error\_records-1].

Type: int16\_t

Default value: -1

#### **ext\_abort\_normal\_wt\_cacheable\_write\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as ext\_abort\_normal\_cacheable\_write\_ras\_type, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `int8_t`

Default value: `-1`

### **`ext_abort_prefetch_device_GRE_read_is_critical`**

Critical reporting of external aborts generated by device-GRE instruction fetches.

Type: `bool`

Default value: `false`

### **`ext_abort_prefetch_device_GRE_read_is_sync`**

Behaviour of external aborts generated by device-GRE instruction fetches. 0, asynchronous abort. 1, synchronous abort. 2, same as `ext_abort_prefetch_is_sync`.

Type: `uint8_t`

Default value: `2`

### **`ext_abort_prefetch_device_nGRE_read_is_critical`**

Critical reporting of external aborts generated by device-nGRE instruction fetches.

Type: `bool`

Default value: `false`

### **`ext_abort_prefetch_device_nGRE_read_is_sync`**

Behaviour of external aborts generated by device-nGRE instruction fetches. 0, asynchronous abort. 1, synchronous abort. 2, same as `ext_abort_prefetch_is_sync`.

Type: `uint8_t`

Default value: `2`

### **`ext_abort_prefetch_device_read_is_critical`**

Critical reporting of external aborts generated by device-nGnRE instruction fetches.

Type: `bool`

Default value: `false`

### **`ext_abort_prefetch_device_read_is_sync`**

Behaviour of external aborts generated by device-nGnRE instruction fetches. 0, asynchronous abort. 1, synchronous abort. 2, same as `ext_abort_prefetch_is_sync`.

Type: `uint8_t`

Default value: `2`

**ext\_abort\_prefetch\_is\_critical**

Critical reporting of external aborts generated by normal writeback cacheable instruction fetches.

Type: `bool`

Default value: `false`

**ext\_abort\_prefetch\_is\_sync**

Behaviour of external aborts generated by normal writeback cacheable instruction fetches. 0, asynchronous abort. 1, synchronous abort.

Type: `bool`

Default value: `true`

**ext\_abort\_prefetch\_noncacheable\_read\_is\_critical**

Critical reporting of external aborts generated by normal noncacheable instruction fetches.

Type: `bool`

Default value: `false`

**ext\_abort\_prefetch\_noncacheable\_read\_is\_sync**

Behaviour of external aborts generated by normal noncacheable instruction fetches. 0, asynchronous abort. 1, synchronous abort. 2, same as `ext_abort_prefetch_is_sync`.

Type: `uint8_t`

Default value: 2

**ext\_abort\_prefetch\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, `number_of_error_records`-1].

Type: `uint16_t`

Default value: `0x0`

**ext\_abort\_prefetch\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `uint8_t`

Default value: 0

**ext\_abort\_prefetch\_so\_read\_is\_critical**

Critical reporting of external aborts generated by device-nGnRnE instruction fetches.

Type: `bool`

Default value: `false`

**ext\_abort\_prefetch\_so\_read\_is\_sync**

Behaviour of external aborts generated by device=nGnRnE instruction fetches. 0, asynchronous abort. 1, synchronous abort. 2, same as `ext_abort_prefetch_is_sync`.

Type: `uint8_t`

Default value: 2

**ext\_abort\_prefetch\_wt\_cacheable\_read\_is\_critical**

Critical reporting of external aborts generated by normal writethrough cacheable instruction fetches.

Type: `bool`

Default value: `false`

**ext\_abort\_prefetch\_wt\_cacheable\_read\_is\_sync**

Behaviour of external aborts generated by normal writethrough cacheable instruction fetches. 0, asynchronous abort. 1, synchronous abort. 2, same as `ext_abort_prefetch_is_sync`.

Type: `uint8_t`

Default value: 2

**ext\_abort\_so\_prefetch\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as `ext_abort_prefetch_ras_index`, Valid indices in range [0, `number_of_error_records`-1].

Type: `int16_t`

Default value: -1

**ext\_abort\_so\_prefetch\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as `ext_abort_prefetch_ras_type`, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `int8_t`

Default value: -1

**ext\_abort\_so\_read\_is\_critical**

Critical reporting of device-nGnRnE read external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_so\_read\_is\_sync**

Synchronous reporting of device-nGnRnE read external aborts.

Type: `bool`

Default value: `true`

**ext\_abort\_so\_read\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number\_of\_error\_records-1].

Type: `uint16_t`

Default value: `0x0`

**ext\_abort\_so\_read\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `uint8_t`

Default value: `0`

**ext\_abort\_so\_write\_is\_critical**

Critical reporting of device-nGnRnE write external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_so\_write\_is\_sync**

Synchronous reporting of device-nGnRnE write external aborts.

Type: `bool`

Default value: `true`

**ext\_abort\_so\_write\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number\_of\_error\_records-1].

Type: `uint16_t`

Default value: `0x0`

#### **`ext_abort_so_write_ras_type`**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `uint8_t`

Default value: `0`

#### **`ext_abort_ttw_cacheable_read_is_critical`**

Critical reporting of TTW cacheable read external aborts.

Type: `bool`

Default value: `false`

#### **`ext_abort_ttw_cacheable_read_is_sync`**

Synchronous reporting of TTW cacheable read external aborts.

Type: `bool`

Default value: `true`

#### **`ext_abort_ttw_cacheable_read_ras_index`**

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number\_of\_error\_records-1].

Type: `uint16_t`

Default value: `0x0`

#### **`ext_abort_ttw_cacheable_read_ras_type`**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `uint8_t`

Default value: `0`

#### **`ext_abort_ttw_noncacheable_read_is_critical`**

Critical reporting of TTW noncacheable read external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_ttw\_noncacheable\_read\_is\_sync**

Synchronous reporting of TTW noncacheable read external aborts.

Type: `bool`

Default value: `true`

**ext\_abort\_ttw\_noncacheable\_read\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number\_of\_error\_records-1].

Type: `uint16_t`

Default value: `0x0`

**ext\_abort\_ttw\_noncacheable\_read\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `uint8_t`

Default value: `0`

**ext\_abort\_ttw\_wt\_cacheable\_read\_is\_critical**

Critical reporting of TTW write-through cacheable read external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_ttw\_wt\_cacheable\_read\_is\_sync**

Synchronous reporting of TTW write-through cacheable read external aborts. 0, asynchronous. 1, synchronous. 2, same as `ext_abort_ttw_cacheable_read_is_sync`.

Type: `uint8_t`

Default value: `2`

**ext\_abort\_ttw\_wt\_cacheable\_read\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as `ext_abort_ttw_cacheable_read_ras_index`, Valid indices in range [0, number\_of\_error\_records-1].

Type: `int16_t`

Default value: `-1`



**ext\_abort\_ttw\_wt\_cacheable\_read\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as ext\_abort\_ttw\_cacheable\_read\_ras\_type, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `int8_t`

Default value: `-1`

**external\_oslar\_access\_disabled\_by\_authentication**

If true, external accesses to OSLAR, when external debugging is not enabled, will generate an error (FEAT\_Debugv8p2).

Type: `bool`

Default value: `false`

**far\_unchanged\_for\_stack\_alignment\_fault**

If true and a Stack Pointer Alignment Fault occurs, FAR\_ELx is not updated, whatever value it had is maintained.

Type: `bool`

Default value: `false`

**far\_unchanged\_for\_virtual\_serror**

If true and a virtual SError occurs, FAR\_ELx is not updated, whatever value it had is maintained.

Type: `bool`

Default value: `false`

**far\_unchanged\_when\_fnv\_set**

If true and ESR\_ELx.FnV=1, FAR\_ELx is not updated but PFAR\_ELx/MFAR\_ELx can be updated.

Type: `bool`

Default value: `false`

**fault\_on\_nT\_bit\_set**

Whether block translation table entries with the nT bit set should always fault. Only applies when changing\_block\_size\_without\_bbm\_support\_level is 1 or higher.

Type: `bool`

Default value: `true`

**fault\_unalign\_to\_unsupported\_access**

If `has_unaligned_single_copy_atomicity` is true, whether unaligned A64 atomic, exclusive and acquire/release instructions to non iWB-oWB or the access crossing a 16-byte boundary generate fault. Bits 0,1,2,3 should be set accordingly to enable the fault behaviour. bit 0: atomic access should fault, bit 1: exclusive access should fault, bit 2: acquire/release should fault, bit 3: the access crossing a 16-byte boundary should fault.

Type: `uint8_t`

Default value: 8

**fault\_unaligned\_s1\_device\_s2\_fwb**

Whether unaligned fault with stage1 Device memory and final memory attribute forced to normal by FWB. 0, No fault. 1, will fault. 2 No fault if final Shareability is NSH.

Type: `uint8_t`

Default value: 0

**flash\_protection\_enable\_at\_reset**

Enable flash memory protection after reset.

Type: `bool`

Default value: `false`

**force\_align\_pc**

**UNPREDICTABLE** branch to non-word-aligned address in ARM state is forced to be aligned.

Type: `bool`

Default value: `false`

**force\_sync\_on\_wfx**

If true, the PE does a context synchronization before entering low power state(WFI/WFE).

Type: `bool`

Default value: `false`

**fpcr\_short\_vector\_raz**

FPSCR and FPCR fields LEN and STRIDE are hardwired to 0.

Type: `bool`

Default value: `false`

**fpsr\_res0\_stateful\_mask**

Mask for stateful bits of FPSR which are **RES0**.

Type: uint32\_t

Default value: 0x0

**gic.GICC-offset**

Offset from PERIPHBASE for GICC registers.

Type: uint32\_t

Default value: 0x2000

**gic.GICD-offset**

Offset from PERIPHBASE for GICD registers. Will be ignored when GICv3 CPU interface is enabled, as distributor is then external to the cluster.

Type: uint32\_t

Default value: 0x1000

**gic.GICH-offset**

Offset from PERIPHBASE for GICH registers.

Type: uint32\_t

Default value: 0x4000

**gic.GICH-other-CPU-offset**

Offset from PERIPHBASE for GICH registers for accessing other CPUs in the cluster. Set to 0 to disable.

Type: uint32\_t

Default value: 0x5000

**gic.GICV-alias**

Offset from PERIPHBASE for alias of GICV registers. When gicv2-only, if zero no alias will be created; if gicv2-only=0, the param is deprecated, when zero or unset an alias is created in the place mandated by the architecture (GICV-base+0xF000).

Type: uint32\_t

Default value: 0x0

**`gic.GICV-offset`**

Offset from PERIPHBASE for GICV registers.

Type: `uint32_t`

Default value: `0x6000`

**`gic.PERIPH-size`**

Size of registers based at PERIPHBASE that are considered to be owned by the GIC. Any accesses in the range PERIPHBASE to PERIPHBASE+gic.PERIPH-size-1 that do not match GIC registers will be treated as **RAZ/WI**.

Type: `uint32_t`

Default value: `0x8000`

**`gic_iri.ARE-fixed-to-one`**

GICv2 compatibility is not supported and GICD\_CTLR.ARE\_\* is always one.

Type: `bool`

Default value: `true`

**`gic_iri.DPG-ARE-only`**

Limit application of DPG bits to interrupt groups for which ARE=1.

Type: `bool`

Default value: `false`

**`gic_iri.DPG-bits-implemented`**

Enable implementation of interrupt group participation bits or DPG bits in GICR\_CTLR.

Type: `bool`

Default value: `false`

**`gic_iri.GICD-alias`**

In GICv2 mode: the base address for a 4k page alias of the first 4k of the Distributor page, in GICv3 mode: the base address of a 64KB page containing message based SPI signalling register aliases(0:Disabled).

Type: `uint64_t`

Default value: `0x0`

**`gic_iri.GICD_ITARGETSR-RAZWI`**

If true, the GICD\_ITARGETS registers are **RAZ/WI**.

Type: `bool`

Default value: `false`

**`gic_iri.GICD_PIDR`**

The value for the GICD\_PIDR registers, if non-zero. Note: fixed fields (device type etc.) will be overridden in this value.

Type: `uint64_t`

Default value: `0x0`

**`gic_iri.GICR_PIDR`**

The value for the GICR\_PIDR registers, if non-zero. Note: fixed fields (device type etc.) will be overridden in this value.

Type: `uint64_t`

Default value: `0x0`

**`gic_iri.GITS_PIDR`**

The value for the GITS\_PIDR registers, if non-zero. Note: fixed fields (device type etc.) will be overridden in this value.

Type: `uint64_t`

Default value: `0x0`

**`gic_iri.ICFGR-rsvd-bit`**

If ARE=0, the value of reserved bits i.e. bit 0,2,4..30 of ICFGRn for n>0.

Type: `bool`

Default value: `true`

**`gic_iri.IIDR`**

GICD\_IIDR and GICR\_IIDR value.

Type: `uint32_t`

Default value: `0x0`

**`gic_iri.IRI-ID-bits`**

Number of bits used to represent interrupts IDs in the Distributor and Redistributors, forced to 10 when none of LPIs, extended SPIs or extended PPIs is supported.

Type: `uint8_t`

Default value: 16

**`gic_iri.ITS-count`**

Number of Interrupt Translation Services to be instantiated (0=none).

Type: `uint8_t`

Default value: 0

**`gic_iri.SPI-count`**

Number of SPIs that are implemented.

Type: `uint16_t`

Default value: 32

**`gic_iri.SPI-message-based-support`**

Distributor supports message based signaling of SPI.

Type: `bool`

Default value: `true`

**`gic_iri.STATUSR-implemented`**

Determines whether the GICR\_STATUSR register is implemented.

Type: `bool`

Default value: `false`

**`gic_iri.enable_protocol_checking`**

Enable/disable protocol checking at cpu interface.

Type: `bool`

Default value: `false`

**`gic_iri.enabled`**

Enable GICv3 functionality; when false the component is inactive.

Type: `bool`

Default value: `true`

### **`gic_iri.has-two-security-states`**

If true, has two security states.

Type: `bool`

Default value: `false`

### **`gic_iri.irouter-default-mask`**

Default Mask value for IROUTER[32..1019] register in the form 'a.b.c.d'.

Type: `string`

Default value: `"0.0.0.7"`

### **`gic_iri.irouter-default-reset`**

Default Reset Value of IROUTER[32..1019] register in the form 'a.b.c.d' or \*.

Type: `string`

Default value: `"0.0.0.0"`

### **`gic_iri.monolithic`**

Indicate that the implementation is not distributed.

Type: `bool`

Default value: `true`

### **`gic_iri.non-ARE-core-count`**

Maximum number of non-ARE cores; normally used to pass the cluster-level NUM\_CORES parameter to the top-level redistributor.

Type: `uint8_t`

Default value: 4

### **`gic_iri.periph-size`**

Size in bytes allocated to internal GIC Distributor.

Type: `uint32_t`

Default value: `0x0`

### **`gic_iri.priority-bits`**

Number of implemented priority bits.

Type: `uint8_t`

Default value: 5

### **`gic_iri.processor-numbers`**

Specify processor numbers (as appears in GICR\_TYPER) in the form 0.0.0.0=0,0.0.0.1=1 etc.) If not specified, will number processors starting at 0.

Type: `string`

Default value: N/A

### **`gic_iri.redistributor-offset`**

Offset from reg-offset where the Redistributors are accessible.

Type: `uint32_t`

Default value: 0x0

### **`gic_iri.redistributor-size`**

Per Redistributor register space in bytes.

Type: `uint32_t`

Default value: 0x0

### **`gic_iri.reg-offset`**

Offset from PERIPHBASE allocated to internal GIC Distributor.

Type: `uint32_t`

Default value: 0x0

### **`gic_iri.supports-shareability`**

Device supports shareability attributes on outgoing memory bus (i.e. is modelling an ACElite port rather than an AXI4 port).

Type: `bool`

Default value: `false`

### **`gic_iri.virtual-lpi-support`**

GICv4 Virtual LPIs and Direct injection of Virtual LPIs supported.

Type: `bool`

Default value: `false`



**`gic_iri.wakeup-on-reset`**

Go against specification and start redistributors in woken-up state at reset. This allows software that was written for previous versions of the GICv3 specification to work correctly. This should not be used for production code or when the distributor is used separately from the core fast model.

Type: `bool`

Default value: `false`

**`gicv3.A3-affinity-supported`**

Whether a non-zero value for affinity at level 3 is supported.

Type: `bool`

Default value: `false`

**`gicv3.BPR-min`**

The minimum value for the GICC\_BPR register (non-secure version will be 1 + this value).

Type: `uint8_t`

Default value: 2

**`gicv3.EOI-check-CPUID`**

Check CPU ID specified for accesses to EOI registers (rather than just ending highest priority active interrupt).

Type: `bool`

Default value: `false`

**`gicv3.EOI-check-ID`**

Check Interrupt ID specified for accesses to EOI registers (rather than just ending highest priority active interrupt).

Type: `bool`

Default value: `false`

**`gicv3.EOI-deactivate-any-interrupt`**

Allow an EOI to deactivate interrupts that aren't the highest priority active interrupt (EOI-ignore-out-of-order must be false otherwise this is ignored).

Type: `bool`

Default value: `false`

**`gicv3.EOI-ignore-out-of-order`**

Ignore EOI writes that cannot end the highest priority active interrupt.

Type: `bool`

Default value: `true`

**`gicv3.FIQEn-RAO`**

GICC\_CTLR.FIQEn is read as one, write insensitive.

Type: `bool`

Default value: `false`

**`gicv3.IIDR_base`**

The base value for calculating the GICC\_IIDR register value.

Type: `uint32_t`

Default value: `0x43b`

**`gicv3.LR-count`**

The number of implemented list registers.

Type: `uint8_t`

Default value: `16`

**`gicv3.PMHE-RAO-WI`**

ICC\_CTLR\_EL\*.PHME is read as one, write insensitive.

Type: `bool`

Default value: `false`

**`gicv3.PMHE-RAZ-WI`**

ICC\_CTLR\_EL\*.PHME is read as zero, write insensitive.

Type: `bool`

Default value: `false`

**`gicv3.PMHE-release-set-packet`**

if PHME is enabled, whether a SET packet is released by CPU Intf in Upstream Ack window.

Type: `bool`

Default value: `false`

**gicv3.SRE-EL2-enable-RAO**

When ICC\_SRE\_EL2.SRE is **RAO/WI**, makes ICC\_SRE\_EL2.Enable **RAO/WI**.

Type: `bool`

Default value: `false`

**gicv3.SRE-EL3-enable-RAO**

When ICC\_SRE\_EL3.SRE is **RAO/WI**, makes ICC\_SRE\_EL3.Enable **RAO/WI**.

Type: `bool`

Default value: `false`

**gicv3.SRE-EL3-set-once**

Restrict SRE EL3 to be set only once.

Type: `bool`

Default value: `false`

**gicv3.SRE-enable-action-on-mmap**

Allowed values are: 0-SRE one allows mmap access. 1-SRE one disables mmap access. 2-SRE one makes mmap access **RAZ-WI**.

Type: `uint8_t`

Default value: `0`

**gicv3.STATUSR-implemented**

If GICv3 CPU interface is being used, this determines whether the STATUS registers are implemented.

Type: `bool`

Default value: `true`

**gicv3.VBPR-min**

The minimum value for the GICV\_BPR register (non-secure version will be 1 + this value).

Type: `uint8_t`

Default value: `2`

**gicv3.VFIQEn-RAO**

ICH\_VMCR\_EL2.VFIQEn is read as one, write insensitive.

Type: `bool`

Default value: `false`

### **`gicv3.cputntf-mmap-access-level`**

Allowed values are: 0-mmap access is supported for GICC,GICH,GICV registers. 1-mmap access is supported only for GICV registers. 2-mmap access is not supported.

Type: `uint8_t`

Default value: `0`

### **`gicv3.dir-trap-support`**

The cpu supports separate trapping of ICC\_DIR\_EL1 to EL2.

Type: `bool`

Default value: `true`

### **`gicv3.el3_trap_priority_when_secure_debug_disabled`**

Undef to access priorities group register when secure debug is disabled.

Type: `bool`

Default value: `false`

### **`gicv3.extended-interrupt-range-support`**

Device has support for extended SPI/PPI ID ranges.

Type: `bool`

Default value: `false`

### **`gicv3.gicv2-only`**

Limit the GIC implementation to GICv2 features only.

Type: `bool`

Default value: `false`

### **`gicv3.idle-is-ff`**

For GICC/GICV RPR, when idle, return FF when true, minimum supported priority otherwise.

Type: `bool`

Default value: `true`

**gicv3.ignore-DIR-write-when-EOImode-not-set**

Ignore **UNPREDICTABLE** access to GICC\_DIR register.

Type: `bool`

Default value: `true`

**gicv3.interrupt-bypass-support**

Interrupt bypass support, set to false for devices not supporting interrupt bypass.

Type: `bool`

Default value: `true`

**gicv3.local-SEIs**

Generate SEI to signal internal issues.

Type: `bool`

Default value: `false`

**gicv3.local-VSEIs**

Generate VSEI to signal internal issues.

Type: `bool`

Default value: `false`

**gicv3.physical-ID-bits**

Number of physical ID bits implemented.

Type: `uint8_t`

Default value: 16

**gicv3.priority-bits**

Number of priority bits implemented.

Type: `uint8_t`

Default value: 5

**gicv3.send-PMHE-command-only-when-priority-changes**

Send PMHE upstream command to distributor only when write to ICC\_PMR\_EL1 changes the priority.

Type: `bool`

Default value: `false`

### **`gicv3.sgi-range-selector-support`**

Device has support for the Range Selector feature for SGI.

Type: `bool`

Default value: `false`

### **`gicv3.suppress-virtual-enables-comms`**

In GICv3 only mode, prevents the GIC CPUIF from communicating UpstreamWrite/VirtualEnables to the IRI.

Type: `bool`

Default value: `true`

### **`gicv3.virtual-ID-bits`**

Number of virtual ID bits implemented.

Type: `uint8_t`

Default value: 16

### **`gicv3.virtual-lpi-support`**

When GICv3 is supported, indicates a cut down CPUIF interface with no support of VLPI (GICv3 only) when false.

Type: `bool`

Default value: `true`

### **`gicv3.virtual-priority-bits`**

Number of virtual priority bits implemented.

Type: `uint8_t`

Default value: 5

### **`gicv3.without-DS-support`**

GICv3 CPU interfaces do not support disabling security in the distributor (GICD\_CTLR.DS=1).

Type: `bool`

Default value: `false`

**`gicv4.mask-virtual-interrupt`**

If true, virtual interrupts can be masked from being reported to virtual CPU interface by setting ICH\_HCR\_EL2.DVIM 1. No control otherwise.

Type: `bool`

Default value: `false`

**`gicv5.config_file`**

File path for the GICv5 configuration yaml. The file lists the GICv5 params.

Type: `string`

Default value: `N/A`

**`gicv5.has_gcie_legacy`**

When set to true, FEAT\_GCIE\_LEGACY is supported.

Type: `bool`

Default value: `false`

**`gicv5.interrupt-bypass-support`**

Interrupt bypass support. when set to true, bypasses GICv5 CPU interface to signal interrupts to the PE.

Type: `bool`

Default value: `false`

**`global_debug_rom.ROMDEVID`**

Value of Debug Rom Device Identification Register.

Type: `uint64_t`

Default value: `0x0`

**`global_debug_rom.ROMPIDR`**

Value of Debug Rom Peripheral Identification Register.

Type: `uint64_t`

Default value: `0x4000bb000`

**`global_debug_rom.ROMPRIDR0`**

Value of Debug ROM Power RequestID Register.

Type: `uint8_t`

Default value: 1

### **hardware\_translation\_table\_update\_implemented**

Implement hardware translation table updates from ARMv8R-64. values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8R-64 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has-gicv4.1**

GICv4.1 is enabled, and all the features with GICv4.1 are implemented (FEAT\_GICv4p1).

Type: `bool`

Default value: `false`

### **has\_16bit\_asids**

Enable 16-bit ASIDs; mandatory in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **has\_16bit\_vmids**

Implement support for 16-bit VMIDs from ARMv8R-64. values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8R-64 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_16k\_granule**

Implement the 16k LPAE translation granule.

Type: `bool`

Default value: `false`

### **has\_4k\_granule**

Implement the 4k LPAE translation granule.

Type: `bool`

Default value: `true`



**has\_64k\_granule**

Implement the 64k LPAE translation granule.

Type: `bool`

Default value: `true`

**has\_aarch32\_dbgdidr\_etc**

DBGDIDR, DBGDRAR, DBGDSAR exist even if EL1 doesn't implement AArch32.

Type: `bool`

Default value: `true`

**has\_aarch64**

All implemented exception levels can run in AArch64.

Type: `bool`

Default value: `false`

**has\_bc**

Implement Armv8.8 Hinted Conditional Branch (FEAT\_HBC) values of this parameter are:- 1, feature is implemented if ARMv8.8 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

**has\_ccidx**

Implement the ARMv8R-64 CCSIDR Extension. Extending the ccsidr number of sets (FEAT\_CCIDX). values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.3 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

**has\_cluster\_l1cache\_size**

Whether core supports cluster level l1cache size.

Type: `bool`

Default value: `true`

**has\_coherent\_icache**

Whether icache invalidation to the point of unification is required for instruction to data coherence.  
true - Invalidate operations not required.

Type: `bool`

Default value: `false`

**has\_const\_pac**

Feature for singular selection of PAC field (FEAT\_CONSTPACFIELD).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8R-64 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

**has\_cvadp\_support**

Implement instruction to support cache clean by deep persistence (DC CVADP) from ARMv8.5, can be selected for core implemented on any arch version starting ARMv8.2 (FEAT\_DPB, FEAT\_DPB2).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.2 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

**has\_debug\_rom**

If true, a debug ROM will be generated describing the cluster's debug components.

Type: `bool`

Default value: `true`

**has\_delayed\_ctireg**

Delay the functional effect of CTI register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**has\_delayed\_mdscr\_el1**

Delay the functional effect of MDSCR\_EL1 register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**has\_delayed\_oslar\_el1**

Delay the functional effect of OSLAR\_EL1 register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**has\_delayed\_pmureg**

Delay the functional effect of PMU register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**has\_dgh**

Implements Data Gathering Hint instruction from ARMv8.6 (FEAT\_DGH).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.6 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

**has\_e0pd**

Implement ARMv8-R64 feature to prevent unprivileged access to one half of the memoryvalues of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8R-64 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

**has\_edacr**

Implement EDACR register.

Type: `bool`

Default value: `true`

**has\_enhanced\_pac**

If pointer authentication is enabled then implement enhanced PAC.

Type: `bool`

Default value: `false`

**has\_exception\_trapping\_form\_of\_vector\_catch**

Implement the exception trapping form of vector catch debug event.

Type: `bool`

Default value: `true`

**has\_export\_m\_port**

The interrupt distributor has an optional interrupt export port for routing interrupts to an external device.

Type: `bool`

Default value: `true`

**has\_far\_not\_valid**

Implements FnV bit in ESR\_ELx and xFSR, FAR not valid for synchronous external aborts.

Type: `bool`

Default value: `false`

**has\_far\_not\_valid\_dfsc**

Implements FnV bit in ESR\_ELx, FAR not valid for synchronous external aborts for Data Abort.

Type: `bool`

Default value: `false`

**has\_far\_not\_valid\_ifsc**

Implements FnV bit in ESR\_ELx and xFSR, FAR not valid for synchronous external aborts for Instruction Abort.

Type: `bool`

Default value: `false`

### **has\_flash**

Flash Port present.

Type: `bool`

Default value: `false`

### **has\_flash\_protection**

Implement flash memory protection.

Type: `bool`

Default value: `false`

### **has\_fp16**

Implement the half-precision floating-point data processing instructions from ARMv8R-64 (FEAT\_FP16).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8R-64 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_generic\_authentication**

Implement ARMv8.3 generic authentication.values of this parameter are:- 1, feature is implemented if ARMv8R-64 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_hardware\_translation\_table\_update**

Type of hardware translation table supported (when enabled by `hardware_translation_table_update_implemented`). 0, not implemented. 1, access bit updates implemented. 2, access bit updates and dirty bit mechanism implemented.

Type: `uint8_t`

Default value: 2

### **has\_internal\_gic\_iri**

Is Internal GIC IRI implemented.

Type: `bool`

Default value: `false`

**has\_itd**

Implement the optional IT disable feature.

Type: `bool`

Default value: `true`

**has\_large\_system\_ext**

Implement the ARMv8 Large System Extensions (FEAT\_LSE).

Type: `bool`

Default value: `false`

**has\_large\_va**

Implement support for the extended 52-bit virtual addresses from ARMv8R-64 (FEAT\_LVA).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8R-64 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

**has\_llpp**

Low Latency Peripheral Port present.

Type: `bool`

Default value: `false`

**has\_mpm**

Implement max-power mitigation mechanism (MPMM).

Type: `bool`

Default value: `false`

**has\_no\_os\_double\_lock**

Do not implement the OS double-lock (FEAT\_DoubleLock).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8R-64 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

**has\_par\_bit10\_razwi**

Whether PAR\_EL1[10] is **RAZ/WI**.

Type: `bool`

Default value: `false`

**has\_partial\_delayed\_mdscr\_el1**

has\_delayed\_oslar\_el1 only apply to some bits of MDSCR\_EL1 (MDE, KDE, TDCC, SS).

Type: `bool`

Default value: `false`

**has\_pc\_sample\_based\_profiling**

If true, pc sample-based profiling is enabled (FEAT\_PCSRv8, FEAT\_PCSRv8p2).

Type: `bool`

Default value: `true`

**has\_per\_cluster\_debug\_auth\_ports**

If true then the debug authentication ports i.e. spniden, niden, rpliden, rtpiden, dbgen, spiden are available per cluster.

Type: `bool`

Default value: `false`

**has\_pl2**

Whether EL2 is implemented.

Type: `bool`

Default value: `true`

**has\_pmc**

Programmable MBIST controllers implemented.

Type: `bool`

Default value: `false`

**has\_pmu**

Implement the optional Performance Monitors Extension (FEAT\_PMUv3). 0, Not Implemented. 1, Implemented. 2, PMU is IMPLEMENTATION\_DEFINED, PMU version would be set to 0xF and would behave as if no PMU is implemented.

Type: `uint8_t`

Default value: 1

### **has\_pointer\_authentication**

Implement ARMv8.3 pointer authentication (FEAT\_PAuth).values of this parameter are:- 1, feature is implemented if ARMv8R-64 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_prediction\_invalidation\_instructions**

Implement execution and data prediction invalidation from ARMv8-R64 (FEAT\_SPECRES).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8R-64 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_qarma3\_pac**

Supports QARMA3 pointer authentication algorithm (FEAT\_PACQARMA3).

Type: `bool`

Default value: `false`

### **has\_ras**

Implements the ARMv8 RAS Extension. 0 = NO\_RAS, 1 = MINIMAL\_RAS, 2 = FULL\_RAS (FEAT\_RAS).

Type: `uint8_t`

Default value: 0

### **has\_ras\_armv84\_extension**

Implement ARMv8R-64 RAS Extension (FEAT\_RASv1p1).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8R-64 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **has\_ras\_double\_fault**

Implement ARMv8.4 RAS Double Fault Extension (FEAT\_DoubleFault).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8R-64 is enabled.- 2, feature is implemented.



Type: `uint8_t`

Default value: 0

### **has\_restriction\_on\_speculative\_data\_loaded**

Implements the ARMv8-R64 security feature (Restrictions on the effects of speculation) (FEAT\_CSV3) values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8R-64 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_self\_hosted\_trace\_extension**

Implement support for the Self-hosted Trace Extensions from ARMv8R-64 (FEAT\_TRF). values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8R-64 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_small\_page\_table**

Implement small page table support which increases the maximum value of TxSZ field from ARMv8R-64 (FEAT\_TTST). Note: will be unimplemented only if both `has_small_page_table=0x0` and `has_pl2=0x0`. values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8R-64 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_software\_lock**

Implement software lock in memory-mapped CTI, PMU, and external debug interfaces.

Type: `bool`

Default value: `true`

### **has\_speculation\_barrier\_inst**

Implement speculation barrier instruction (SB) from ARMv8-R64 (FEAT\_SB). values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8R-64 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

**has\_speculative\_sei**

If true, the PE can generate SError interrupt exceptions from speculative reads of memory, including speculative instruction fetches.

Type: `bool`

Default value: `false`

**has\_spp**

Shared Peripheral Port present.

Type: `bool`

Default value: `false`

**has\_stage2\_ap\_speculative\_update**

Speculative update of S2 AP bit on S1 TTW. 0 = No Update, 1 = Update, 2 = Update including for AT ops.

Type: `uint8_t`

Default value: 0

**has\_synchronous\_load\_atomics**

Report asynchronous abort due to unsupported load atomics as synchronous (Cacheable).

Type: `bool`

Default value: `true`

**has\_synchronous\_load\_atomics\_noncacheable**

Report asynchronous abort due to unsupported load atomics as synchronous (Non-Cacheable).

Type: `bool`

Default value: `true`

**has\_synchronous\_store\_atomics**

Report asynchronous abort due to unsupported store atomics as synchronous (Cacheable).

Type: `bool`

Default value: `false`

**has\_synchronous\_store\_atomics\_noncacheable**

Report asynchronous abort due to unsupported store atomics as synchronous (Non-Cacheable).

Type: `bool`

Default value: `false`

### **has\_tlb\_conflict\_abort**

Detected inconsistent TLB content generate aborts. Ignored if FEAT\_BBML3 is implemented.

Type: `bool`

Default value: `false`

### **has\_tlb\_pa\_caching**

Whether intermediate caching of translation table walks might include NonCoherent caches of previous valid walks. 0, NonCoherent caches might be included. 1, No NonCoherent caches included (FEAT\_nTLBPA).

Type: `bool`

Default value: `false`

### **has\_unsupported\_exclusive\_fault**

Report unsupported exclusive access with Unsupported Exclusive fault status (otherwise use external abort).

Type: `bool`

Default value: `true`

### **has\_v8\_4\_pmu\_extension**

Implement PMU extension from ARMv8.4 (FEAT\_PMUv3p4).values of this parameter are:- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_v8\_5\_debug\_over\_power\_down**

Implement ARMv8.5 Debug over powerdownvalues of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8R-64 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **has\_v8\_6\_pmu\_events**

Implements PMU events from ARMv8.6values of this parameter are:- 1, feature is implemented if ARMv8.6 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_writebuffer**

Implement write accesses buffering before L1 cache. May affect `ext_abort` behaviour.

Type: `bool`

Default value: `false`

### **hcptr\_tta\_behaviour**

Behaviour of HCPTR.TTA when there is no CP14 ETM interface. 0, **RAZ/WI**. 1, **RAO/WI**. 2, stateful.

Type: `uint8_t`

Default value: 2

### **hcr\_el2\_miocnce\_is\_rw**

If true, HCR\_EL2.MIOCNCE is treated as R/W instead of **RAZ/WI**; always set to false in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **hcr\_swio\_res1**

Whether HCR.SWIO and/or HCR\_EL2.SWIO are **RES1**.

Type: `bool`

Default value: `false`

### **hsr\_uncond\_cc**

Condition codes reported in HSR as AL if it passes.

Type: `bool`

Default value: `false`

### **icache-hit\_latency**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-log2linelen**

If nonzero, Log2 of the instruction cache line length in bytes (valid values in range 4-8). Otherwise the value of cache-log2linelen is used.

Type: `uint8_t`

Default value: 0

**icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

Type: `uint64_t`

Default value: 0x0

**icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

Type: `uint64_t`

Default value: 0x0

**icache-nprefetch**

Number of next sequential instruction cache lines to prefetch. This is only used when icache-prefetch\_enabled=true.

Type: `uint32_t`

Default value: 0x1

**icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

Type: `bool`

Default value: false

**icache-prefetch\_level**

0 based cache level at which instructions are pre-fetched. This is only used when icache-prefetch\_enabled=true.

Type: `uint8_t`

Default value: 0

**icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-read\_bus\_width\_in\_bytes**

L1 I-Cache read bus width in bytes used to calculate per-access timing annotations.

Type: `uint32_t`

Default value: `0x8`

**icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-size**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: `0x8000`

**icache-state\_modelled**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

**icache-ways**

L1 I-Cache number of ways (sets are implicit from size).

Type: `uint8_t`

Default value: `2`

**ignore\_tag\_check\_dcc\_load\_store\_in\_ma\_mode\_when\_tco\_is\_disabled**

Constrained unpredictable behavior for reads/writes to external debug interface DTR regs in memory access mode when PSTATE.TCO is 0. If true, tag check is ignored else, tag check is performed if required.

Type: `bool`

Default value: `false`

**imp\_def\_functionality\_behaviour**

Behaviour of **IMPLEMENTATION DEFINED** registers and system instructions. 0, UNDEF. 1, **RAZ/WI**.

Type: `uint8_t`

Default value: 0

**inner\_cache\_boundary**

CLIDR.ICB, cache level boundary between inner and outer shareable domains.

Type: `uint8_t`

Default value: 0

**instruction\_tlb\_size**

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

Type: `uint32_t`

Default value: 0x0

**internal\_vgic**

Instantiate VGIC peripheral in this processor.

Type: `bool`

Default value: `false`

**is\_debug\_state\_pmu\_snapshot\_allowed**

If true, PMU snapshot is allowed in debug state.

Type: `bool`

Default value: `true`

**is\_first\_pcsr\_sample\_ignored**

If true, First read of PMPCSR register after reset returns 0xFFFFFFFF.

Type: `bool`

Default value: `false`

### **`is_ras_irq_edge_triggered`**

If true, ras interrupt is edge-triggered. Otherwise, it's level-triggered.

Type: `bool`

Default value: `true`

### **`is_serror_edge_triggered`**

If true, SError is edge-triggered. Otherwise, it's level-triggered.

Type: `bool`

Default value: `true`

### **`is_uniprocessor`**

Value for the U bit in MPIDR. true disables L1 cache coherency protocols.

Type: `bool`

Default value: `false`

### **`isb_is_branch`**

If true, ISB is considered an immediate branch. This allows to count ISB as a branch in BRBE.

Type: `bool`

Default value: `false`

### **`ish_is_osh`**

Whether Innershareable is same as OuterShareable.

Type: `bool`

Default value: `false`

### **`itd_conditional_instructions_are_32bit`**

When SCTLR\_ELx.ITD=1, an IT instruction plus a T16 instruction are considered a single 32bit conditional instruction.

Type: `bool`

Default value: `false`



**`jidr_is_undef_at_el0`**

If true, JIDR register access is UNDEF at EL0.

Type: `bool`

Default value: `false`

**`jmcr_is_undef_at_el0`**

If true, JMCR register access is UNDEF at EL0.

Type: `bool`

Default value: `false`

**`joscr_is_undef_at_el0`**

If true, JOSCR register access is UNDEF at EL0.

Type: `bool`

Default value: `false`

**`l1cache_has_r52_cache_policy`**

Whether L1cache has r52 cache policy.

Type: `bool`

Default value: `false`

**`l1cache_has_rsvd_flash_ways`**

Whether L1 cache segregates ways for flash and AXI data.

Type: `bool`

Default value: `false`

**`l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**`l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l2cache-read_bus_width_in_bytes`**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: `uint32_t`

Default value: `0x8`

### **`l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

**l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: `uint8_t`

Default value: `16`

**l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: `uint32_t`

Default value: `0x8`

**l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-has\_mpam**

L3 Cache has MPAM support.

Type: `bool`

Default value: `false`

**l3cache-hit\_latency**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-maintenance\_latency**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-miss\_latency**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-mpamf.arch\_major\_ver**

L3 Cache MPAMF\_AIDR architecture major version.

Type: `uint8_t`

Default value: `0`

**l3cache-mpamf.arch\_minor\_ver**

L3 Cache MPAMF\_AIDR architecture minor version.

Type: `uint8_t`

Default value: `0`

**l3cache-mpamf.esr\_mask**

L3 Cache MPAMF\_ESR mask value.

Type: `uint32_t`

Default value: `0xffffffff`

### **`l3cache-mpamf.has_esr`**

L3 Cache's MPAMF\_ESR, MPAMF\_ECR, and MPAM error handling implemented.

Type: `bool`

Default value: `false`

### **`l3cache-mpamf.has_extd_esr`**

L3 Cache's MPAMF\_ESR is 64-bits.

Type: `bool`

Default value: `false`

### **`l3cache-mpamf.has_impl_idr`**

L3 Cache's MPAMF\_IMPL\_IDR is present.

Type: `bool`

Default value: `false`

### **`l3cache-mpamf.has_mbwu_long_counter`**

L3 Cache has long MBWU counter and capture registers.

Type: `bool`

Default value: `false`

### **`l3cache-mpamf.has_mpamfidr_ext`**

MPAMF\_IDR.EXT support.

Type: `bool`

Default value: `false`

### **`l3cache-mpamf.has_partid_nrw`**

Narrowing part ID register is present. This is global rather than per-instance.

Type: `bool`

Default value: `false`

### **`l3cache-mpamf.has_priority_partitioning`**

The selected resource has priority partitioning described in MPAMF\_PRI\_IDR.

Type: `bool`

Default value: `false`

### **`l3cache-mpamf.has_prod_id`**

L3 Cache MPAMF\_IIDR product ID supported.

Type: `uint16_t`

Default value: `0x0`

### **`l3cache-mpamf.has_prod_rev`**

L3 Cache MPAMF\_IIDR product REVISION supported.

Type: `uint8_t`

Default value: `0`

### **`l3cache-mpamf.has_prod_var`**

L3 Cache MPAMF\_IIDR product VARIENT supported.

Type: `uint8_t`

Default value: `0`

### **`l3cache-mpamf.has_ris`**

L3 Cache has resource instance selection support.

Type: `bool`

Default value: `false`

### **`l3cache-mpamf.max_partid_ns`**

L3 Cache Maximum value of non-secure PARTID supported.

Type: `uint16_t`

Default value: `0xffff`

### **`l3cache-mpamf.max_partid_rl`**

L3 Cache Maximum value of realm PARTID supported for RME implementations.

Type: `uint16_t`

Default value: `0xffff`

### **`l3cache-mpamf.max_partid_rt`**

L3 Cache Maximum value of root PARTID supported for RME implementations.

Type: `uint16_t`

Default value: `0xffff`

### **`l3cache-mpamf.max_partid_s`**

L3 Cache Maximum value of secure PARTID supported.

Type: `uint16_t`

Default value: `0xffff`

### **`l3cache-mpamf.max_pmg_ns`**

L3 Cache Maximum value of non-secure PMG supported.

Type: `uint8_t`

Default value: 255

### **`l3cache-mpamf.max_pmg_rl`**

L3 Cache Maximum value of realm PMG supported for RME implementations.

Type: `uint8_t`

Default value: 255

### **`l3cache-mpamf.max_pmg_rt`**

L3 Cache Maximum value of root PMG supported for RME implementations.

Type: `uint8_t`

Default value: 255

### **`l3cache-mpamf.max_pmg_s`**

L3 Cache Maximum value of secure PMG supported.

Type: `uint8_t`

Default value: 255

### **`l3cache-mpamf.mbwu_long_counter_width`**

L3 Cache long MBWU counter width. 0: 63 bits, 1: 44 bits.

Type: `uint8_t`

Default value: 0

### **`l3cache-mpamf.no_impl_msmon`**

L3 Cache's MPAMF\_IMPL\_IDR does not describe resource monitors.

Type: `bool`

Default value: `false`

### **`l3cache-mpamf.no_impl_part`**

L3 Cache's MPAMF\_IMPL\_IDR does not describe resource partitioning controls.

Type: `bool`

Default value: `false`

### **`l3cache-mpamf.ris_max`**

L3 Cache's largest resource instance selector value defined.

Type: `uint8_t`

Default value: `0`

### **`l3cache-mpamf_base`**

L3 Cache memory mapped MPAM registers base address.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-read_access_latency`**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size `l3cache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-read_bus_width_in_bytes`**

L3 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: `uint32_t`

Default value: `0x8`

### **`l3cache-read_latency`**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l3cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.



Type: `uint64_t`

Default value: `0x0`

### **l3cache-size**

L3 Cache size in bytes.

Type: `uint32_t`

Default value: `0x0`

### **l3cache-snoop\_data\_transfer\_latency**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **l3cache-snoop\_issue\_latency**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **l3cache-ways**

L3 Cache number of ways (sets are implicit from size).

Type: `uint16_t`

Default value: `16`

### **l3cache-write\_access\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size `l3cache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **l3cache-write\_bus\_width\_in\_bytes**

L3 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: `uint32_t`

Default value: 0x8

### **l3cache-write\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. l3cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **legacy\_combining\_exc\_catch\_trace**

Whether exception catch is traced as part of exception entry/exit in same cycle.

Type: bool

Default value: true

### **limit\_ete\_revision\_without\_rme**

If true, limit the ETE revision to ETEv1p1 when FEAT\_RME is not implemented.

Type: bool

Default value: false

### **ls64\_ignore\_s1\_unpred\_memattr\_transformation**

If true, stage 1 unpredictable memory attribute transformations are ignored for FEAT\_LS64 single-copy atomic 64-byte load/store instructions' (FEAT\_LS64, FEAT\_LS64\_V, FEAT\_LS64\_ACCDATA).

Type: bool

Default value: false

### **ls64\_memtype\_check\_use\_combined\_memattr**

FEAT\_LS64 single-copy atomic 64-byte load/store instructions' 0 : memory attributes check is performed at each enabled stage of translation, 1 : memory attributes check is done on the combined memory attributes only. 2. memory attributes check is done on the combined memory attributes with Stage1 and Stage2 fault get evaluated to check on which stage fault should be reported.

Type: uint8\_t

Default value: 0

### **ls64wb\_memtype\_check\_allow\_any\_cacheable\_memattr**

If true, when FEAT\_LS64WB is implemented, any cacheable memory access performed by LD/ST64B instructions is 64-byte, single-copy atomic.

Type: `bool`

Default value: `false`

### **mdrar\_el1\_res0**

MDRAR\_EL1 is **RES0**.

Type: `bool`

Default value: `false`

### **memory.ext\_slave\_base**

Base address of Slave Port. Each core's region will offset by `ext_slave_size_per_core`.

Type: `uint64_t`

Default value: `0x0`

### **memory.ext\_slave\_size\_per\_core**

Size of Slave region for each core.

Type: `uint32_t`

Default value: `0x0`

### **memory.flash\_base**

Base address of Flash.

Type: `uint32_t`

Default value: `0x0`

### **memory.flash\_size**

Size of the Flash RAM.

Type: `uint32_t`

Default value: `0x0`

### **memory.has\_llram**

Low-Latency RAM present.

Type: `bool`

Default value: `false`

### **memory.l2\_cache.is\_inner\_cacheable**

L2 cache obeys inner cacheable attributes (rather than outer cacheable attributes).

Type: `bool`

Default value: `true`

### **`memory.l2_cache.is_inner_shareable`**

L2 cache obeys inner shareable attributes (rather than outer shareable attributes).

Type: `bool`

Default value: `true`

### **`memory.llram_base`**

Base address of LLRAM.

Type: `uint64_t`

Default value: `0x0`

### **`memory.llram_enable_at_reset`**

Whether llram is enabled at reset.

Type: `bool`

Default value: `false`

### **`memory.llram_shared`**

Controls the Low-Latency RAM's sharability attribute.

Type: `bool`

Default value: `false`

### **`memory.llram_size`**

Size of the LLRAM.

Type: `uint32_t`

Default value: `0x0`

### **`memory.scu_present`**

L1 Caches are coherent.

Type: `bool`

Default value: `true`

### **`memory.transmit_vmid_in_user_flags`**

Transmit VMID in transaction attributes.

Type: `bool`

Default value: `false`

### **`mixed_endian`**

Implement support for mixed endianness. 0, not supported. 1, supported at all exception levels. 2, supported at ELO only. Unsupported in the presence of Future Architecture Technologies (FAT).

Type: `uint8_t`

Default value: 1

### **`mpidr_layout`**

Layout of MPIDR. 0 AFF0 is CPUID, 1 AFF1 is CPUID.

Type: `uint8_t`

Default value: 0

### **`non_secure_vgic_alias_when_ns_only`**

If ! `has_el3` and only non-secure side exists, then the normal position of the VGIC is a secure alias. If this parameter is non-zero then in addition a non-secure alias of the VGIC will be placed at this position (aligned to 32 KB).

Type: `uint64_t`

Default value: `0x0`

### **`num_protection_regions_s1`**

Number of v8-R protection regions.

Type: `uint8_t`

Default value: 32

### **`num_protection_regions_s2`**

Number of v8-R hyp protection regions.

Type: `uint8_t`

Default value: 32

### **`num_spi`**

Number of interrupts (SPI) into the internal GIC controller.

Type: `uint16_t`

Default value: 32

**number\_of\_error\_records**

Cores Number of Error records supported for RAS.

Type: `uint16_t`

Default value: `0x0`

**page\_based\_hardware\_attributes**

Implement the page based hardware attributes from ARMv8R-64. This parameter indicates which page table bits are available for hardware, where bits[3:0] correspond to PTE[62:59] and to TCR\_ELx.HWUnyy (FEAT\_HPDS2).

Type: `uint8_t`

Default value: `0`

**par\_ns\_set\_unknown\_bit**

Whether NS bit of PAR is set/clear when executing AT to perform non-secure regime translation. When true, NS is set to 1 else 0.

Type: `bool`

Default value: `true`

**par\_nse\_set\_unknown\_bit**

Whether NSE bit of PAR is set/clear when executing AT operation on secure, non-secure or realm translation regime. When true, NSE is set to 1 else 0.

Type: `bool`

Default value: `false`

**per\_core\_master\_supported**

If master port from each core is exposed out of cluster.

Type: `bool`

Default value: `false`

**pfr1\_csv2\_frac**

Fractional revision number ID\_AA64PFR1\_EL1.CSV2\_frac when ID\_AA64PFR0\_EL1.CSV==1 for CSV2 extension (FEAT\_CSV2\_1p1, FEAT\_CSV2\_1p2).

Type: `uint8_t`

Default value: `0`

**plbi\_invalid\_xt**

If true, PLBI instructions with invalid Xt are treated as UNDEF.

Type: `bool`

Default value: `false`

**pmb\_idr\_external\_abort**

Describes how the PE manages External aborts on writes made by the Statistical Profiling Extension to the Profiling Buffer. 0, External abort is reported to SPE, From Armv8.8 and Armv9.3, the value 0 is not permitted. 1, External abort is ignored. 2, The External abort generates an SError and the error is not reported to SPE.

Type: `uint8_t`

Default value: 0

**pmb\_idr\_flag\_updates**

Defines whether the address translation performed by the Profiling Buffer manages the Access Flag and dirty state.

Type: `bool`

Default value: `true`

**pmbsr\_dl\_razwi**

Whether PMBSR\_ELx.DL is **RAZ/WI** or behaves as specified, indicating partial loss of a record due to a buffer management event or external abort.

Type: `bool`

Default value: `false`

**pmbsr\_ea\_razwi**

Whether PMBSR\_ELx.EA is **RAZ/WI** or set as the result of an external abort.

Type: `bool`

Default value: `false`

**pmcr\_disable\_events\_export**

If true, export for PMU events is disabled. This configures PMCFGR.EX field.

Type: `bool`

Default value: `true`

**pmmir\_el1\_bus\_slots**

Largest value by which BUS\_ACCESS can increment over BUS\_CYCLES cycles.

Type: uint16\_t

Default value: 0

**pmmir\_el1\_bus\_width**

Width, in bytes, of accesses counted by BUS\_ACCESS.

Type: uint16\_t

Default value: 0x0

**pms\_idr\_max\_size**

Defines largest size for a single SPE record (rounded up to a power of 2).

Type: uint8\_t

Default value: 6

**pmu-num\_counters**

Number of PMU counters implemented.

Type: uint8\_t

Default value: 8

**pmu\_cycle\_counter\_counts\_actual\_cycles**

If true and Timing annotation is enabled, PMU cycle counter counts actual cycles, otherwise counts instructions executed.

Type: bool

Default value: false

**pmu\_has\_chain\_event**

PMU (if present) implements event number 0x1e, CHAIN.

Type: bool

Default value: true

**preserve\_cache\_contents\_over\_warm\_reset**

Preserve cache contents over warm reset by ignoring the value of ram\_clear\_on\_reset\_disable signal in the cache.



Type: `bool`

Default value: `false`

### **pseudo\_fault\_generation\_feature\_register**

ARMv8.4 Standard Pseudo-fault generation feature register values. JSON schema for the parameter value is: [{"OF":false, "UC":false, "UEU":false, "UER":false, "UEO":false, "DE":false, "CE":0x0, "CI":false, "ER":false, "PN":false, "AV":false, "MV":false, "SYN":false, "R":false, "NA":false}, other\_pseudo-fault\_generating\_features\_register\_values]. Where OF, UC, UEU, UER, UEO, DE, CI, ER, PN, AV, MV, SYN, and R have valid false(NOT\_SUPPORTED) and true(FEATURE\_CONTROLLABLE), where CE can have 0(NOT\_SUPPORTED), 1(NONSPECIFIC\_CE\_SUPPORTED) and 3(TRANSIENT\_OR\_PERSISTENT\_CE\_SUPPORTED) and NA can have false(component fakes detection on next access) or true(component fakes detection spontaneously). Effective only when ERXFR's INJ field allows it or has\_ras\_fault\_injection is true.

Type: `string`

Default value: `N/A`

### **pstate\_ssbs\_type**

Implement speculative store bypass safe feature from ARMv8.5. 0, Not supported. 1, Supported without MSR/MRS access to SSBS (FEAT\_SSBS). 2, fully supported (FEAT\_SSBS2).

Type: `uint8_t`

Default value: `0`

### **ptw\_latency**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

### **ram\_protection\_enable\_at\_reset**

Enable TCM, L1Cache memory protection after reset.

Type: `bool`

Default value: `false`

### **randomize\_unknowns\_at\_reset**

Will fill in unknown bits in registers at reset with random value using register\_reset\_data as seed, it overrides scramble\_unknowns\_at\_reset.

Type: `bool`

Default value: `false`

**ras\_extra\_configurations**

Miscellaneous configurations for error records. An array of JSON objects. Note for ERXCTLR\_EL1 register it only allows to define the mask value for the IMPDEF fields, ie bits [63:32] and bit 1, but its reset value applies on all fields. Note for ERXMISCN masks - these are 64 bit masks covering the 64 bit registers ERXMISCN\_EL1. E.g. [{"Index": 0, "ERXMISCO\_mask": 0x0, "ERXMISCO\_reset": 0x0, "ERXMISC1\_mask": 0x0, "ERXMISC1\_reset": 0x0, "ERXMISC2\_mask": 0x0, "ERXMISC2\_reset": 0x0, "ERXMISC3\_mask": 0x0, "ERXMISC3\_reset": 0x0, "ERXCTLR\_EL1\_mask": 0x0, "ERXCTLR\_EL1\_reset": 0x0}, {"Index": 1, "ERXMISCO\_mask": 0x0, "ERXMISCO\_reset": 0x0, "ERXSTATUS\_IERR\_mask": 0x300}].

Type: `string`

Default value: `N/A`

**ras\_pfg\_clock\_mhz**

RAS Pseudo-Fault generation clock rate in MHz.

Type: `uint8_t`

Default value: `24`

**register\_reset\_data**

Data used to fill register bits when they become **UNKNOWN** at reset.

Type: `uint64_t`

Default value: `0x0`

**register\_reset\_data\_hi**

Data used to fill the upper-half of 128-bit registers when the bits become **UNKNOWN** at reset.

Type: `uint64_t`

Default value: `0x0`

**report\_iside\_cmo\_ifsr**

fault info for an iside cache maintenance operation is reported in the IFSR.

Type: `bool`

Default value: `true`

**report\_second\_access\_align\_fault\_non\_atomic\_pair\_access**

If true, the non-atomic load/store pair accesses report the 2nd register as faulting address for an alignment fault. This is IMP-DEF behavior as defined in FEAT\_LRCPC3.

Type: `bool`

Default value: `false`

### **`report_second_access_mmu_fault_non_atomic_pair_access`**

If true, the non-atomic load/store pair accesses report the 2nd register as faulting address for an MMU fault. This is IMP-DEF behavior as defined in FEAT\_LRCPC3.

Type: `bool`

Default value: `false`

### **`reported_fp_revision`**

Purely cosmetic patch level value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored. Updates the FPSID.revision value.

Type: `int8_t`

Default value: -1

### **`reported_patch_level`**

Purely cosmetic patch level value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: `int8_t`

Default value: -1

### **`reported_revision_number`**

Purely cosmetic revision number value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: `int8_t`

Default value: -1

### **`reserved_HMC_SSC_PAC_treated_disabled`**

When DBG[B|W]CR.{HMC,SSC,PAC} bits configuration is reserved, this parameter controls whether breakpoints/watchpoints are treated as Disabled or not.

Type: `bool`

Default value: `false`

### **`restore_fpsr_on_trapped_fp_exception`**

If true, FPSR is restored to the value of the FPSR immediately before the instruction that generated the trapped floating-point exception.

Type: `bool`

Default value: `false`

### **`restriction_on_speculative_execution`**

Implements the ARMv8-R64 security feature (Restrictions on the effects of speculation): 0: No disclosure whether branch targets trained in one context can affect speculative execution in a different context, 1: Branch targets trained in one context cannot affect speculative execution in a different hardware described context (SCXTNUM\_ELx not supported), 2: Branch targets trained in one context cannot affect speculative execution in a different hardware described context (SCXTNUM\_ELx supported) (FEAT\_CSV2, FEAT\_CSV2\_2).

Type: `uint8_t`

Default value: 0

### **`rmr_always_implemented`**

Always implement RMR\_ELx, RMR, or HRMR at the highest implemented exception level, even if that exception level cannot use both AArch32 and AArch64.

Type: `bool`

Default value: `false`

### **`s1_align_memtype_fault_prio_more_than_s2_perm_fault_on_s1_walk`**

If true, s1 alignment fault has priority over s2 permission faults.

Type: `bool`

Default value: `true`

### **`s1_perm_fault_prio_more_than_s2_perm_fault_on_s1_walk`**

If true, s1 permission fault has priority over s2 on s1 translation table walk permission faults.

Type: `bool`

Default value: `false`

### **`s1_unsupported_atomic_fault_for_ls64_prio_more_than_s2_perm_fault`**

If true, unsupported atomic/exclusive faults due to LS64 instructions at Stage 1 have higher priority than permission fault at Stage 2.

Type: `bool`

Default value: `false`

### **`scheduler_mode`**

Control the interleaving of instructions in this processor. 0, default long quantum. 1, low latency mode, short quantum and signal checking. 2, lock-breaking mode, long quantum with additional

context switches near load-exclusive instructions. WARNING: This parameter is intended for validation purposes and may result in unwanted behaviour if altered!.

Type: `uint8_t`

Default value: 0

#### **`scr_nET_writeable`**

Whether SCR.nET is writeable. Writing to it is purely cosmetic (nET behavior not implemented).

Type: `bool`

Default value: `false`

#### **`scramble_unknowns_at_reset`**

Will fill in unknown bits in registers at reset with `register_reset_data`.

Type: `bool`

Default value: `true`

#### **`seerror_clear_delay`**

Delay for clearing of SError if SError is level-triggered, in cpu cycles.

Type: `uint32_t`

Default value: `0x0`

#### **`skip_trace_on_write_to_oseccr_el1_when_oslock_is_unlocked`**

If `OSLSR_EL1.OSLK == 0`, then `OSECCR_EL1` returns an unknown value on reads and ignores writes. When true, also skips the traces on writes to `OSECCR_EL1` when `OSLSR_EL1.OSLK == 0`.

Type: `bool`

Default value: `false`

#### **`spp.base`**

Sets the base address of Shared Peripheral Port.

Type: `uint64_t`

Default value: `0x0`

#### **`spp.size`**

Sets the size of SPP(in bytes).

Type: `uint32_t`

Default value: 0x1000

**spsr\_el3\_is\_mapped\_to\_spsr\_mon**

Whether SPSR\_EL3 is mapped to AArch32 register SPSR\_mon.

Type: bool

Default value: false

**spsr\_m4\_res0**

Whether SPSR\_ELx.M[4] bit should be **RES0** for AARCH64 only implementations.

Type: bool

Default value: false

**stage12\_tlb\_size**

If VMSA is supported at stage1, number of stage1+2 tlb entries. If instruction\_tlb\_size !=0, this is treated as dtlb size.

Type: uint32\_t

Default value: 0x0

**stage1\_tlb\_size**

Number of stage1 only tlb entries. Valid values are 0 or >= 4.

Type: uint32\_t

Default value: 0x0

**stage1\_walkcache\_size**

Number of stage1 only walk cache entries.

Type: uint32\_t

Default value: 0x0

**strex\_fail\_can\_hit\_watchpoint**

If true, a strex fail can hit watchpoint.

Type: bool

Default value: false

**supports\_multi\_threading**

Sets the MPIDR.MT bit. Setting this to true hints the the cluster is multi-threading compatible.

Type: `bool`

Default value: `false`

### **`swp_with_xzr_is_st_atomic`**

If true, swp with dest as xzr is treated as store atomic.

Type: `bool`

Default value: `true`

### **`take_ccfail_tsc_trap`**

When `take_ccfail_undef=1` this parameter controls whether or not an SMC instruction that is trapped by HCR\_EL2.TSC but fails its condition code check generates a trap to EL2.

Type: `bool`

Default value: `false`

### **`take_ccfail_undef`**

UNDEF exception is taken even if condition code check fails.

Type: `bool`

Default value: `true`

### **`tcr_tgx_bit1_stateful`**

TCR.TGx[1] is stateful even without 16k granule support.

Type: `bool`

Default value: `false`

### **`tidcp_traps_el0_undef_imp_def`**

TIDCP has priority over UNDEF for accesses to **IMPLEMENTATION DEFINED** functionality from EL0.

Type: `bool`

Default value: `true`

### **`tlb_latency`**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

**tlbi\_or\_ic\_invalid\_xt**

Behavior of TLBI and IC instructions that don't take Xt as an argument when Xt != 0b11111. 0: TLB and IC not UNDEF, 1: TLBI UNDEF, IC not UNDEF, 2: TLBI not UNDEF, IC UNDEF, 3: TLBI and IC UNDEF.

Type: `uint8_t`

Default value: 0

**trace\_has\_sysreg\_access**

ETM trace registers support access via system registers.

Type: `bool`

Default value: `true`

**trace\_icc\_registers\_as\_icv\_when\_redirected**

If true, update trace with ICV, instead of ICC when ICV registers are accessed depending on the core state.

Type: `bool`

Default value: `false`

**trace\_physical\_registers\_when\_host\_virtualisation\_enabled**

Trace sysreg accesses to physical register accessed (0=disabled, 1=Trace only ELR/SPSR\_EL1 as ELR/SPSR\_EL2, 2=Trace all redirected registers as physical registers affecting all features inc. the host virtualised registers.

Type: `uint8_t`

Default value: 0

**trace\_xzr\_in\_core\_regs64\_trace**

Whether CORE\_REGS64\_READ traces XZR and WZR input registers.

Type: `bool`

Default value: `true`

**trap\_dc\_cmo\_to\_pou\_if\_nop**

Whether traps to DC CMO operations to PoU are ignored if the same is treated as **NOP**.

Type: `bool`

Default value: `true`



**trap\_ic\_cmo\_to\_pou\_if\_nop**

Whether traps to IC CMO operations to PoU are ignored if the same is treated as **NOP**.

Type: `bool`

Default value: `true`

**trap\_reserved\_group3\_id\_regs**

Whether setting HCR\_EL2.TID3 traps reserved group3 id registers.

Type: `bool`

Default value: `false`

**treat-dcache-cmos-to-poc-as-nop**

Whether dcache maintenance operations to the point of coherency are required for instruction to data coherence. 0 - Clean/Invalidate ops required, 1 - Clean/Invalidate ops not required and cannot generate faults, 2 - Clean/Invalidate ops not required but can generate faults.

Type: `uint8_t`

Default value: 0

**treat-dcache-cmos-to-pou-as-nop**

Whether dcache invalidation to the point of unification is required for instruction to data coherence. 0 - Invalidate ops required, 1 - Invalidate ops not required and cannot generate faults, 2 - Invalidate ops not required but can generate faults.

Type: `uint8_t`

Default value: 0

**treat-dcache-invalidate-as-clean-invalidate**

Treat data cache invalidate operations as clean and invalidate.

Type: `bool`

Default value: `false`

**treat-icache-cmos-to-pou-as-nop**

If `has_coherent_icache` is true, whether instruction cache invalidation operations to PoU which are treated as **NOP** can generate fault. 0 - cannot generate faults, 1 - can generate faults.

Type: `uint8_t`

Default value: 0

**`treat_forced_normal_as_device_for_excl_atomics`**

Whether exclusive/atomic access is supported in same manner as access to device if stage1 is Device memory and final memory attribute forced to normal by FWB.

Type: `bool`

Default value: `false`

**`treat_pld_as_nop`**

If true, treat PLD as **NOP**.

Type: `bool`

Default value: `false`

**`treat_pli_as_nop`**

If true, treat PLI as **NOP**.

Type: `bool`

Default value: `false`

**`treat_wfi_wfe_as_nop`**

If true, never go into wait state for WFI or WFE instructions.

Type: `bool`

Default value: `false`

**`truncate_pc_on_illegal_exception_return_to_aarch32`**

On Illegal ERET to AArch32, truncate PC to 32-bits.

Type: `bool`

Default value: `true`

**`unification-level`**

Level of Unification Inner Shareable for the cache hierarchy.

Type: `uint8_t`

Default value: `1`

**`unification-uniprocessor-level`**

Level of Unification Uniprocessor for the cache hierarchy.

Type: `uint8_t`

Default value: 1

**unpred\_LSE128\_overlap**

Constrained unpredictable behaviours for 128-bit LSE overlap. 1 Constraint\_UNDEF, 2 Constraint\_NOP.

Type: uint8\_t

Default value: 1

**unpred\_brbe\_next\_branch\_cycle\_count\_unknown**

If true, cycle count value for the next BRBE branch record after BRB INJ execution outside prohibited region is unknown.

Type: bool

Default value: false

**unpred\_clear\_ISV\_for\_exception\_before\_software\_step**

Whether ESR\_ELx.ISV bit is cleared/set, when it is constrained unpredictable due to a different exception before a software step exception.

Type: bool

Default value: false

**unpred\_edscr\_ns\_set\_unknown\_bit**

Unknown(x) bit in NS field in EDSCR can be configure to 0 or 1.

Type: bool

Default value: false

**unpred\_edscr\_rw\_unknown\_bits\_read\_as\_1**

Unknown(x) bits in RW field in EDSCR are read as 1 instead of 0.

Type: bool

Default value: false

**unpred\_edscr\_status\_read\_as\_no\_syndrome**

Controls the choice of EDSCR.STATUS bit-values, when it is constrained unpredictable behaviour due to a different exception before a halting step debug event.

Type: bool

Default value: false

**unpred\_extdbg\_unknown\_bits**

Data used to fill only in **UNKNOWN** bit-fields of external debug registers e.g., EDPFR and EDDFR.

Type: `uint64_t`

Default value: `0x0`

**unpred\_load\_single\_reg\_overlap\_with\_wb**

Constrained unpredictable behaviours for single load with writeback(might impact certain load pair instructions) 0 Constraint\_WBSUPPRESS, 1 Constraint\_UNDEF, 2 Constraint\_NOP.

Type: `uint8_t`

Default value: `0`

**unpred\_mrsmsr\_currentlymapped\_undef**

**UNPREDICTABLE** register access (accessible from current mode using different instruction) modeled as **NOP** when false and **UNDEF** when true.

Type: `bool`

Default value: `false`

**unpred\_mrsmsr\_protfailed\_undef**

**UNPREDICTABLE** register access (not accessible from current PL and security state) modeled as **NOP** when false and **UNDEF** when true.

Type: `bool`

Default value: `false`

**unpred\_par\_attr\_returns\_mair**

If true, `PAR_EL1.ATTR` represents the memory attributes as per the MAIR value instead of the ones in the descriptor.

Type: `bool`

Default value: `false`

**unpred\_s2\_hw\_dirty\_update\_on\_atomic\_wo\_read\_perm\_fault**

Constrained unpredictable behavior for atomic instructions that generate a stage 2 permission fault only due to lack of read permission, on a stage 2 writable-clean descriptor. If true, hardware is allowed to update the stage 2 dirty state; else, the dirty update is suppressed.

Type: `bool`

Default value: `false`

**unpred\_sctlr\_c\_0\_taggable\_behaviour**

Controls unpredictable effects when SCTLTR\_ELx.C=0 for a stage 1 translation regime on whether memory is treated as taggable. Values: 0=Tagged, 1=Untagged but forced to Tagged when FWB=1 and stage 2 restores WB, 2 = Untagged.

Type: uint8\_t

Default value: 2

**unpred\_stage2\_mpu\_and\_bg\_disabled**

Constrained unpredictable when stage2 MPU and background disabled. 0, Stage-2 level 0 translation fault(Default). 1, Unknown memory attributes.

Type: uint8\_t

Default value: 0

**unpred\_store\_exclusive\_base\_overlap**

Constrained unpredictable behaviours for store exclusive when s==n. 0 Constraint\_NONE, 1 Constraint\_UNDEF, 2 Constraint\_NOP.

Type: uint8\_t

Default value: 0

**unpred\_store\_pair\_and\_single\_reg\_overlap\_with\_wb**

Constrained unpredictable behaviours for pair and single store with writeback(doesn't cover store exclusive) 0 Constraint\_NONE, 1 Constraint\_UNDEF, 2 Constraint\_NOP.

Type: uint8\_t

Default value: 0

**unpred\_tchange\_tenter\_and\_texit\_behaviour**

TCHANGE, TENTER and TEXTIT unpredictable behaviour in debug state. 0, **NOP**. 1, Undefined. 2, Execute as in non-debug state.

Type: uint8\_t

Default value: 0

**unpred\_tlbi\_not\_in\_monitor\_mode**

Constrained unpredictable behaviors for AArch32 TLBI instructions executed in secure privileged mode other than Monitor mode. 0: Preferred behavior (default), 1: UNDEF, 2: **NOP**, 3: execute as if had been executed in Monitor mode.

Type: uint8\_t

Default value: 0

### **unpred\_tsize\_aborts**

Behaviour when TSize is out of range. 0, force into range. 1, translation fault, forces unpred\_tsize\_pamax\_aborts to 1.

Type: `bool`

Default value: `false`

### **unpred\_tsize\_pamax\_aborts**

Behaviour when stage 2 TSize exceeds the physical address size (or 40bits, from AArch32). 0, force into range. 1, translation fault. Ignored if unpred\_tsize\_aborts is 1.

Type: `bool`

Default value: `false`

### **unpredictable\_exclusive\_abort\_memtype**

Cause MMU abort if exclusive access is not supported in certain memory type (0=exclusives allowed in all memory types, 1=exclusives abort in Device memory types, 2=exclusives abort in any type other than WB inner cacheable).

Type: `uint8_t`

Default value: 0

### **unpredictable\_hvc\_behaviour**

HVC unpredictable behaviour. 0, UNDEF. 1, **NOP**.

Type: `uint8_t`

Default value: 0

### **unpredictable\_smc\_behaviour**

SMC unpredictable behaviour. 0, UNDEF. 1, **NOP**.

Type: `uint8_t`

Default value: 0

### **unpredictable\_wfet\_and\_wfit\_behaviour**

WFET and WFIT unpredictable behaviour in debug state. 0, **UNDEFINED**. 1, **NOP**.

Type: `uint8_t`

Default value: 1

**unsupported\_atomic\_fault\_type**

Type of fault reported on unsupported atomic access. 0 = external abort if any reported by interconnect, 1 = precise unsupported atomic fault, 2 = precise external abort, 3 = imprecise external abort.

Type: `uint8_t`

Default value: 0

**unsupported\_hw\_update\_fault\_type**

Type of abort reported when hw update to descriptor is done using unsupported memtype (0=No abort, 1=IMPDEF abort caused by memtype, 2=Sync external abort).

Type: `uint8_t`

Default value: 0

**use\_architectural\_names**

Use names SP/LR/PC instead of R13/R14/R15.

Type: `bool`

Default value: `false`

**use\_stage1\_sh\_as\_input\_to\_stage2**

IMPDEF case of whether to use stage1 shareability or OuterShareable as input to stage2 if stage1 is Device memtype.

Type: `bool`

Default value: `false`

**use\_tlb\_contig\_hint**

Translation table entries with the contiguous hint bit set generate large TLB entries.

Type: `bool`

Default value: `false`

**user\_defined\_rom\_table\_debug\_power\_config**

User defined ROM Table debug power domains for ED,CTI,PMU and TRACE, and DBGPCR configuration. The "version" field and "cores" array are mandatory. The "dbgpcr" array, if provided, must contain unique integers in the range [0, 31] describing which debug power domains have power control implemented. The "rom" and "dbgpcr" fields in objects in the "cores" array are only allowed when 'debug\_rom\_is\_flat' is false. All power domain ID fields ("rom", "ed/pmu", "cti", "etm") must be in the range [0, 31]. The "ed/pmu" field is mandatory. Example JSON for a hierarchical

debug ROM layout: {"version": 0, "dbgpcr": [0, 1], "cores":[{"dbgpcr": [1, 31], "rom": 0, "ed/pmu": 0, "cti": 31, "etm": 1}, {"ed/pmu": 0}]}

Type: `string`

Default value: `N/A`

### **vpu\_datapath\_width**

VPU data path width.

Type: `uint8_t`

Default value: `128`

### **walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

### **warn\_unpredictable\_in\_v7**

If true, behaviour which is unpredictable in V7 yet is predictable in V8 will produce a warning.

Type: `bool`

Default value: `true`

### **watchpoint-log2secondary\_restriction**

log2 size of secondary restriction of FAR/EDWAR possible values on watchpoint hit for load/store operations.

Type: `uint8_t`

Default value: `0`

### **wfe\_wakeup\_delay**

Configure WFE wakeup delay in CPU cycles.

Type: `uint32_t`

Default value: `0x0`

### **wfi\_wakeup\_delay**

Configure WFI wakeup delay in CPU cycles.

Type: `uint32_t`



Default value: 0x0

### **wp\_ignores\_dbm\_update**

If true, dbm update is ignored on watchpoint hit.

Type: bool

Default value: false

## 3.3 AEMvACT

Defined in `LISA/AEMvACT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following parameters were added:

- `ISV_set_to_0_for_page_boundary_crossing`
- `disable_bti_effects`
- `ext_abort_in_virtual_tag_reads_is_sync`
- `ext_abort_in_virtual_tag_writes_is_sync`
- `far_unchanged_for_stack_alignment_fault`
- `far_unchanged_for_virtual_serror`
- `has_afgdt_virtualisation`
- `has_virtual_tag_check_on_load`
- `inner_cache_boundary`
- `limit_ete_revision_without_rme`
- `mpam_at_ops_use_target_el`
- `pan_applies_before_oa_space_checks`
- `plb_dpot_enabled`
- `plb_irt_enabled`
- `plb_ttt_enabled`
- `plbi_invalid_xt`
- `poe2_mmu_fault_far_aligned`

- `preserve_cache_contents_over_warm_reset`
- `ras_pfg_err_opt`
- `tcr_tgx_bit1_stateful`
- `trbe_trbptr_el1_has_res0`

The following parameters were removed:

- `ete.COMMTRANS`
- `has_tme`
- `tme_disable-read-write-set-optimizations`
- `tme_imp-failures-can-retry`
- `tme_implementation-type`
- `tme_random-memory-access-fail-chance`
- `tme_read-set-size`
- `tme_support-only-guaranteed-mem-attr`
- `tme_tcommit-fails-transactions`
- `tme_wakeup-from-wfe-always-fails-transactions`
- `tme_wfe-fails-transactions`
- `tme_write-set-size`

## About AEMvACT

ARM AEM A-Profile(MP) CPU component - number of cores configurable at runtime.

## Iris and MTI instances for AEMvACT

This model has the following Iris instances:

Name	Instance type
AEMvACT	Cluster_ARM_AEM-A_MP
AEMvACT.AMU	PVBusLogger
AEMvACT.AMU.mapper	PVBusMapper
AEMvACT.DAP	PVBusLogger
AEMvACT.DAP.mapper	PVBusMapper
AEMvACT.DSU	DSU
AEMvACT.DSU.mpam_busslave	PVBusSlave
AEMvACT.MMAP	PVBusLogger
AEMvACT.MMAP.mapper	PVBusMapper
AEMvACT.RAS	PVBusLogger
AEMvACT.RAS.mapper	PVBusMapper
AEMvACT.acp_mapper	PVBusMapper
AEMvACT.cpu0	ARM_AEM-A_MP

Name	Instance type
AEMvACT.cpu0.UTLB	TLB
AEMvACT.cpu0.debug_rom	debug_rom
AEMvACT.cpu0.dtlb	TLB
AEMvACT.cpu0.l1dcache	PVCache
AEMvACT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
AEMvACT.cpu0.l1dcache.upstream[0]	PVBusSlave
AEMvACT.cpu0.l1icache	PVCache
AEMvACT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
AEMvACT.cpu0.l1icache.upstream[0]	PVBusSlave
AEMvACT.ext_bus	PVBusLogger
AEMvACT.ext_bus.mapper	PVBusMapper
AEMvACT.gic_cpuif_decoder_cluster	GLCv3CPUInterfaceDecoder
AEMvACT.global_debug_rom	debug_rom
AEMvACT.l2_cache	PVCache
AEMvACT.l2_cache.downstream[0].pvbusmaster	PVBusMaster
AEMvACT.l2_cache.upstream[Y] (where Y = 0-16)	PVBusSlave
AEMvACT.l2_flusher	AsyncCacheFlushUnit
AEMvACT.secondary_debug_rom	debug_rom

This model has the following MTI trace components:

Name	Component type
AEMvACT.AMU	PVBusLogger
AEMvACT.AMU.mapper	PVBusMapper
AEMvACT.DAP	PVBusLogger
AEMvACT.DAP.mapper	PVBusMapper
AEMvACT.DSU	DSU
AEMvACT.DSU.mpam_busslave	PVBusSlave
AEMvACT.MMAP	PVBusLogger
AEMvACT.MMAP.mapper	PVBusMapper
AEMvACT.RAS	PVBusLogger
AEMvACT.RAS.mapper	PVBusMapper
AEMvACT.acp_mapper	PVBusMapper
AEMvACT.cpu0	ARM_AEM-A_MP
AEMvACT.cpu0.UTLB	TLB
AEMvACT.cpu0.l1dcache	PVCache
AEMvACT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
AEMvACT.cpu0.l1dcache.upstream[0]	PVBusSlave
AEMvACT.cpu0.l1icache	PVCache
AEMvACT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
AEMvACT.cpu0.l1icache.upstream[0]	PVBusSlave

Name	Component type
AEMvACT.ext_bus	PVBusLogger
AEMvACT.ext_bus.mapper	PVBusMapper
AEMvACT.gic_cpuif_decoder_cluster	GLCv3CPUInterfaceDecoder
AEMvACT.l2_cache	PVCache
AEMvACT.l2_cache.downstream[0].pvbusmaster	PVBusMaster
AEMvACT.l2_cache.upstream[Y] (where Y = 0-16)	PVBusSlave
AEMvACT.l2_flusher	AsyncCacheFlushUnit

## Ports for AEMvACT

Port	Direction	Protocol	Description
acp_s	slave	PVBus	AXI ACP slave port.
broadcastcachemaint	slave	Signal	ACE defined pins.
broadcasttinner	slave	Signal	ACE defined pins.
broadcastouter	slave	Signal	ACE defined pins.
cfgend	slave	Signal	This signal is for EE bit initialisation.
cfgsdisable	slave	Signal	This signal disables write access to some secure Interrupt Controller registers.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
clrexmonack	master	Signal	Acknowledge handshake signal for the clrexmonreq signal
clrexmonreq	slave	Signal	Signals the clearing of an external global exclusive monitor
clusterid	slave	Value	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set ManagerID64 of the core.
CNTHPIRQ	master	Signal	Timer signals to SOC.
CNTHPSIRQ	master	Signal	Timer signals to SOC.
CNTHVIRQ	master	Signal	Timer signals to SOC.
CNTHVSIRQ	master	Signal	Timer signals to SOC.
CNTPNSIRQ	master	Signal	Timer signals to SOC.
CNTPSIRQ	master	Signal	Timer signals to SOC.
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.
CNTVIRQ	master	Signal	Timer signals to SOC.
commirq	master	Signal	Interrupt signal from debug communications channel.
config64	slave	Signal	Register width after reset.

Port	Direction	Protocol	Description
cp15sdisable	slave	Signal	This signal disables write access to some system control processor registers.
cpuporeset	slave	Signal	CPU power on reset. Initializes all the processor logic, including debug logic.
CRITICALIRQ	master	Signal	RAS Critical Error Interrupt.
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti0extin	slave	Signal	CTI trace inputs for core 0.
cti0extout	master	Signal	CTI trace outputs for core 0.
cti1extin	slave	Signal	CTI trace inputs for core 1.
cti1extout	master	Signal	CTI trace outputs for core 1.
cti2extin	slave	Signal	CTI trace inputs for core 2.
cti2extout	master	Signal	CTI trace outputs for core 2.
cti3extin	slave	Signal	CTI trace inputs for core 3.
cti3extout	master	Signal	CTI trace outputs for core 3.
cti4extin	slave	Signal	CTI trace inputs for core 4.
cti4extout	master	Signal	CTI trace outputs for core 4.
cti5extin	slave	Signal	CTI trace inputs for core 5.
cti5extout	master	Signal	CTI trace outputs for core 5.
cti6extin	slave	Signal	CTI trace inputs for core 6.
cti6extout	master	Signal	CTI trace outputs for core 6.
cti7extin	slave	Signal	CTI trace inputs for core 7.
cti7extout	master	Signal	CTI trace outputs for core 7.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
ctidbgirq	master	Signal	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	slave	Signal	External debug interface.
dbgnopwrdown	master	Signal	Debug no power down request.
dbgpwrdownack	master	Signal	Debug power down acknowledge.
dbgpwrdownreq	slave	Signal	Debug power down request.
dbgpwrupreq	master	Signal	Debug power up request.
dev_debug_s	slave	PVBus	External debug interface.
ERRORIRQ	master	Signal	RAS Error Recovery Interrupt.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
external_trace_reset	slave	Signal	ETMv4 External Trace Reset signal.
FAULTIRQ	master	Signal	RAS Fault Handling Interrupt
fiq_nmi	slave	Signal	-
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s	slave	GICv3Comms	GICv3 cpu interface ports.
hacdbsirg	master	Signal	Interrupt signal from the HACDBS unit.
irq_nmi	slave	Signal	-

Port	Direction	Protocol	Description
irq	slave	Signal	This signal drives the CPUs interrupt handling.
irqs	slave	Signal	These signals drive the CPU's interrupt controller interrupt lines.
l2reset	slave	Signal	This signal resets timer and interrupt controller.
memorymapped_amu_s	slave	PVBus	External interface for amu.
memorymapped_debug_s	slave	PVBus	External debug interface.
niden	slave	Signal	External debug interface.
periphbase	slave	Value_64	This port sets the base address of private peripheral region.
pmbirq	master	Signal	Interrupt signal from the statistical profiling unit.
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
pmusnapshotacks	master	Signal	-
pmusnapshotreqs	slave	Signal	-
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	master	PVBus	The core will generate bus requests on this port.
rei	slave	Signal	Individual processor RAM Error Interrupt signal input.
reset	slave	Signal	Raising this signal will put the core into reset mode.
rlpiden	slave	Signal	External debug interface.
romaddr	slave	Value_64	Debug ROM base address.
romaddrv	slave	Signal	Debug ROM base address valid.
rtpiden	slave	Signal	External debug interface.
rvbar	slave	Value_64	Reset vector base address.
sei	slave	Signal	Per core System Error physical pins.
smpnamp	master	Signal	This signals AMP or SMP mode for each core.
spiden	slave	Signal	External debug interface.
spniden	slave	Signal	External debug interface.
standbywfe	master	Signal	This signal indicates if a core is in WFE state.
standbywfi	master	Signal	This signal indicates if a core is in WFI state.
standbywfil2	master	Signal	This signal indicated all cores and L2 are in a power down state
teinit	slave	Signal	This signal provides default exception handling state.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trace_unit_reset	slave	Signal	ETMv4 Trace Unit Reset signal.
trbirq	master	Signal	Interrupt signal from the trace buffer unit.
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfiq	slave	Signal	Virtual FIQ input. Note that the irq/fiq pins are wired directly to the core if there is no internal VGIC. If there is an internal VGIC then these are ignored.
vinithi	slave	Signal	This signal controls of the location of the exception vectors at reset.

Port	Direction	Protocol	Description
virq	slave	Signal	Virtual IRQ input. Note that the irq/fiq pins are wired directly to the core if there is no internal VGIC. If there is an internal VGIC then these are ignored.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Processor Virtual System Error Interrupt request.

## Parameters for AEMvACT

### **cpuX.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **cpuX.CONFIG64**

Register width configuration at reset. 0, AArch32. 1, AArch64.

Type: `bool`

Default value: `true`

### **cpuX.CP15SDISABLE**

Initialize to disable access to some CP15 registers.

Type: `bool`

Default value: `false`

### **cpuX.CP15SDISABLE2**

Initialize to disable access to some CP15 registers (FEAT\_CP15SDISABLE2).

Type: `bool`

Default value: `false`

### **cpuX.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**cpuX.DCZID-log2-block-size**

Log2 of the block size in words cleared by DC ZVA instruction (as read from DCZID\_ELO).

Type: `uint8_t`

Default value: `8`

**cpuX.DCZVA\_single\_write**

Execute the DCZVA as a single write.

Type: `bool`

Default value: `false`

**cpuX.MPIDR-override**

Override of MPIDR value. If nonzero will override the MT, cluster and CPU ID bits in MPIDR.

Type: `uint64_t`

Default value: `0x0`

**cpuX.RVBAR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**cpuX.RVBAR32**

Reset vector address in AARCH32 when VINITHI is not set and `ignore_rvbar_in_aarch32` is set.

Type: `uint32_t`

Default value: `0x0`

**cpuX.SMPnAMP**

Enable broadcast messages necessary for correct SMP operation at reset.

Type: `bool`

Default value: `true`

**cpuX.TEINIT**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`



**cpuX.VINITHI**

Reset value of SCTLR.V.

Type: `bool`

Default value: `false`

**cpuX.aarch32\_reset\_from\_impdef\_addr**

If PE resets into AArch32, Whether execution starts from IMPDEF address or hi/low vector.

Type: `bool`

Default value: `true`

**cpuX.ase-present**

Set whether the model has been built with NEON support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**cpuX.clock\_divider**

Clock divider ratio for asymmetric MP clocking.

Type: `uint32_t`

Default value: `0x1`

**cpuX.clock\_multiplier**

Clock divider ratio for asymmetric MP clocking.

Type: `uint32_t`

Default value: `0x1`

**cpuX.crypto\_aes**

AES instructions supported (requires CryptoPlugin to be loaded). 0, not implemented. 2, AES and PMULL instructions implemented (FEAT\_AES, FEAT\_PMULL).

Type: `uint8_t`

Default value: `2`

**cpuX.crypto\_sha1**

SHA-1 instructions supported (requires CryptoPlugin to be loaded). 0, not implemented. 1, SHA1 instructions implemented (FEAT\_SHA1).

Type: `uint8_t`

Default value: 1

### **`cpuX.crypto_sha256`**

SHA-256 instructions supported (requires CryptoPlugin to be loaded). 0, not implemented. 1, SHA256 instructions implemented (FEAT\_SHA256).

Type: `uint8_t`

Default value: 1

### **`cpuX.crypto_sha3`**

Implement ARMv8.4 SHA-3 instructions (requires CryptoPlugin to be loaded) (FEAT\_SHA3).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **`cpuX.crypto_sha512`**

Implement ARMv8.4 SHA-512 instructions (requires CryptoPlugin to be loaded) (FEAT\_SHA512).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **`cpuX.crypto_sm3`**

Implement ARMv8.4 SM-3 instructions (requires CryptoPlugin to be loaded) (FEAT\_SM3).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **`cpuX.crypto_sm4`**

Implement ARMv8.4 SM-4 instructions (requires CryptoPlugin to be loaded) (FEAT\_SM4).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

**cpuX.cti-intack\_mask**

Set bits represent that the corresponding trigger requires software acknowledge via CTIINTACK.

Type: `uint8_t`

Default value: 1

**cpuX.cti-number\_of\_claim\_bits**

Number of implemented bits in CTICLAIMSET.

Type: `uint8_t`

Default value: 0

**cpuX.cti-number\_of\_triggers**

Number of cti event triggers (default: 8, valid values: {3-32}).

Type: `uint8_t`

Default value: 8

**cpuX.enable\_crc32**

CRC32 instructions supported. 0, not implemented. 1, CRC32 instructions implemented (FEAT\_CRC32).

Type: `uint8_t`

Default value: 0

**cpuX.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**cpuX.etm-present**

Set whether the model has ETM support.

Type: `bool`

Default value: `true`

**cpuX.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `false`

**cpuX.force-fpsid-value**

Value to override the FPSID value to.

Type: `uint32_t`

Default value: `0x0`

**cpuX.has\_hcptr\_tase**

If false, HCPTR.TASE is **RES0**.

Type: `bool`

Default value: `true`

**cpuX.highest-index-of-context-breakpoints**

Highest index of breakpoints that are context aware.

Type: `uint8_t`

Default value: `15`

**cpuX.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

**cpuX.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

**cpuX.number-of-breakpoints**

Number of breakpoints.

Type: `uint8_t`

Default value: `16`

**cpuX.number-of-context-breakpoints**

Number of breakpoints that are context aware.

Type: `uint8_t`

Default value: 16

### **`cpuX.number-of-watchpoints`**

Number of watchpoints.

Type: `uint8_t`

Default value: 16

### **`cpuX.operation_bandwidth`**

Operation width for ARMv8.4 PMU extension.

Type: `uint8_t`

Default value: 1

### **`cpuX.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

### **`cpuX.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

### **`cpuX.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: 0x123456

### **`cpuX.semihosting-T32_HLT`**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

### **`cpuX.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

### **`cpuX.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **`cpuX.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **`cpuX.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`cpuX.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`cpuX.semihosting-prefix`**

Prefix semihosting output with target instance name.

Type: `bool`

Default value: `false`

### **`cpuX.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`cpuX.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`cpuX.semihosting-stderr_istty`**

Result for semihost istty call when argument is stderr.

Type: `bool`

Default value: `true`

### **`cpuX.semihosting-stdin_istty`**

Result for semihost istty call when argument is stdin.

Type: `bool`

Default value: `true`

### **`cpuX.semihosting-stdout_istty`**

Result for semihost istty call when argument is stdout.

Type: `bool`

Default value: `true`

### **`cpuX.semihosting-use_stderr`**

Send stderr from the simulated process to host stderr.

Type: `bool`

Default value: `false`

### **`cpuX.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**cpuX.unpredictable\_WPMASKANDBAS**

Constrained unpredictable handling of watchpoints when mask and BAS fields specified. 0, IGNOREMASK. 1, IGNOREBAS (default). 2, REPEATBAS8. 3, REPEATBAS.

Type: `uint8_t`

Default value: 1

**cpuX.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**cpuX.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**cpuX.vfp-traps**

Implement support for trapping floating-point exceptions.

Type: `bool`

Default value: `true`

**cpuX.vfp-traps-show-all**

Report all trapped floating-point exceptions in the syndrome when a combination occurs.

Type: `bool`

Default value: `false`

**cpuX.wfet\_early\_or\_delayed\_timeout**

WFET early or delayed timeout beyond the threshold value of CNTVCT\_ELO in percentage.

Type: `int8_t`

Default value: 0

**cpuX.wfit\_early\_or\_delayed\_timeout**

WFIT early or delayed timeout beyond the threshold value of CNTVCT\_ELO in percentage.



Type: `int8_t`

Default value: `0`

### **ADFSR-AIFSR-implemented**

ADFSR and AIFSR are implemented.

Type: `bool`

Default value: `false`

### **AIDR**

Value of AIDR\_EL1 register.

Type: `uint64_t`

Default value: `0x0`

### **AMIIDR**

Value of AMU Implementation Identification Register.

Type: `uint64_t`

Default value: `0x43b`

### **AMPIDR**

Value of AMU Peripheral Identification Register.

Type: `uint64_t`

Default value: `0x4000bb000`

### **BPIMVA\_causes\_translation\_lookup**

Do a translation when BPIMVA instruction is executed (which may cause a translation fault).

Type: `bool`

Default value: `false`

### **BROADCASTCACHEMAIN**

Enable broadcasting of cache maintenance operations to downstream caches. The `broadcastcachemaint` signal will override this value if used.

Type: `bool`

Default value: `true`

**BROADCASTINNER**

Enable broadcasting of Inner Shareable transactions. The broadcastinner signal will override this value if used.

Type: `bool`

Default value: `true`

**BROADCASTOUTER**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

Type: `bool`

Default value: `true`

**CCSIDR-L1D\_override**

If nonzero, override the value presented in CCSIDR for L1D (this is cosmetic and does not affect cache behaviour).

Type: `uint64_t`

Default value: `0x0`

**CCSIDR-L1I\_override**

If nonzero, override the value presented in CCSIDR for L1I (this is cosmetic and does not affect cache behaviour).

Type: `uint64_t`

Default value: `0x0`

**CCSIDR-L2\_override**

If nonzero, override the value presented in CCSIDR for L2 (this is cosmetic and does not affect cache behaviour).

Type: `uint64_t`

Default value: `0x0`

**CCSIDR-L3\_override**

If nonzero, allow L3 selection in CSSELR and present this value in CCSIDR (this is cosmetic and does not affect cache behaviour).

Type: `uint64_t`

Default value: `0x0`

**CLUSTER\_ID**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the ManagerID64 of the core.

Type: uint32\_t

Default value: 0x0

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. This value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain events even when cache state modelling is disabled. Possible values are:- 0 = All CMOs are broadcast if architecturally required- 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required.

Type: uint8\_t

Default value: 1

**CPUCFR**

Value of CPU Configuration Register.

Type: uint64\_t

Default value: 0x0

**CTIPIDR**

If non-zero, override the CTI Peripheral Identification Register.

Type: uint64\_t

Default value: 0x0

**CTR-L1Ip-override**

If non-zero, override the L1Ip bits in CTR/CTR\_ELO system register. This does not change the behaviour of the cache, only what is present in the CTR register.

Type: uint8\_t

Default value: 0

**DBGBCR\_BT\_applies\_RES0\_before\_valid\_check**

If true, **RES0** behaviour is applied to DBGBCR(EL1).BT before checking for reserved values for this field.

Type: `bool`

Default value: `true`

**DBGPIDR**

If non-zero, override the Debug Peripheral Identification Register.

Type: `uint64_t`

Default value: `0x0`

**DBGROMADDR**

Initialization value of DBGDRAR register. Bits[55:12] of this register specify the ROM table physical address.

Type: `uint64_t`

Default value: `0x0`

**DBGROMADDRV**

If true, set bits[1:0] of the CP15 DBGDRAR to indicate that the address is valid.

Type: `bool`

Default value: `false`

**DCZID-TBS-log2-allocation-tags-block-size**

Log2 of the block size in words written by a DC ZGBVA or DC GBVA instruction (for FEAT\_MTETC).

Type: `uint8_t`

Default value: `9`

**ERRIIDR**

Value of RAS Implementation Identification Register.

Type: `uint64_t`

Default value: `0xd800143b`

**ERRPIDR**

Value of RAS Peripheral Identification Register.

Type: `uint64_t`

Default value: `0x4100bbd80`

### **ERXMISCO\_mask**

Write Mask for ERXMISCO RAS Register.

Type: `uint64_t`

Default value: `0x0`

### **GICDISABLE**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`

Default value: `true`

### **GMID-log2-block-size**

Log2 of the block size in words accessed by STGM/LDGM/STZGM instructions.

Type: `uint8_t`

Default value: `4`

### **ISV\_set\_to\_0\_for\_page\_boundary\_crossing**

Whether ESR\_ELx.ISV is cleared when a faulted access crosses a page boundary and FAR\_ELx reports the subsequent page.

Type: `bool`

Default value: `false`

### **ISV\_set\_to\_0\_for\_stage2\_synch\_external\_abort**

Whether ESR\_EL2.ISV is set to 0 on stage 2 synchronous external aborts.

Type: `bool`

Default value: `false`

### **MIDR**

Value of MIDR\_EL1 register.

Type: `uint32_t`

Default value: `0x410fd0f0`

**NUM\_CORES**

Number of cores per cluster.

Type: `uint8_t`

Default value: 1

**PA\_SIZE**

Physical address range supported For ARMv8.0 and ARMv8.1 this is limited to 48 bits (FEAT\_LPA).

Type: `uint8_t`

Default value: 40

**PERIPHBASE**

Base address of peripheral memory space.

Type: `uint64_t`

Default value: 0x13080000

**PMCEID0**

Performance Monitor Common Event ID Reg 0 value - 64 bit.

Type: `uint64_t`

Default value: 0xffffffff

**PMCEID1**

Performance Monitor Common Event ID Reg 1 value - 64 bit.

Type: `uint64_t`

Default value: 0xffffffff

**PMSIDR.ArchInst**

Defines whether architecture instruction sampling is implemented or not, if not only micro op sampling is implemented. Model only supports architecture instruction sampling, but allows ID register field to be configured.

Type: `bool`

Default value: `true`

**PMSIDR.CRR**

Defines whether call return branch records (FEAT\_SPE\_CRR) is implemented or not.

Type: `bool`

Default value: `false`

### **PMSIDR.LDS**

Defines whether data source for sampled load instruction is implemented or not. Model does not implement loaded data source, but allows ID register field to be configured.

Type: `bool`

Default value: `false`

### **PMUPIDR**

If non-zero, override the PMU Peripheral Identification Register.

Type: `uint64_t`

Default value: `0x0`

### **VAL\_disable\_slpoe2\_scr\_el3\_traps**

Disable traps from SCR\_EL3 for FEAT\_S1POE2, FEAT\_TPS and FEAT\_TPSP. WARNING: this is a temporary parameter to aid with software support for these traps. It will be removed in the future.

Type: `bool`

Default value: `false`

### **abort\_execution\_from\_device\_memory**

Execution from device memory generates a prefetch abort.

Type: `bool`

Default value: `false`

### **abort\_slpoe2\_fetch\_from\_device\_memory**

Abort S1POE2 table fetches from device memory.

Type: `bool`

Default value: `false`

### **advsimd\_bf16\_support\_level**

Implement BFloat16 operations from ARMv8.6. AArch64 Advanced SIMD and FP BFloat16 instructions are automatically enabled when `has_arm_v8-6` is true.- 0, Not implemented.- 1, AArch64 Advanced SIMD and FP BFloat16 instructions only (FEAT\_BF16).- 2, AArch32 Advanced SIMD and VFP BFloat16 instructions only (FEAT\_AA32BF16).- 3, Both AArch64 Advanced SIMD and FP and AArch32 Advanced SIMD and VFP BFloat16 instructions.

Type: `uint8_t`

Default value: 0

### **`advsimd_i8mm_support_level`**

Implement Int8 matrix multiply operations from ARMv8.6. AArch64 Advanced SIMD and FP Int8 matrix multiply instructions are automatically enabled when `has_arm_v8-6` is true.- 0, Not implemented.- 1, AArch64 Advanced SIMD and FP Int8 matrix multiply instructions only (FEAT\_I8MM).- 2, AArch32 Advanced SIMD and VFP Int8 matrix multiply instructions only (FEAT\_AA32I8MM).- 3, Both AArch64 Advanced SIMD and FP and AArch32 Advanced SIMD and VFP Int8 matrix multiply instructions (FEAT\_I8MM, FEAT\_AA32I8MM).

Type: `uint8_t`

Default value: 0

### **`advsimd_overread`**

AdvSIMD element load operations access all bytes of a 16-byte aligned window, even in Device memory.

Type: `bool`

Default value: `false`

### **`align_pc_on_branch_to_unaligned_pc_aarch32`**

Force PC align for branches to an unaligned PC counter in A32 state.

Type: `bool`

Default value: `false`

### **`align_pc_on_debug_exit_to_aarch32`**

Exit to AARCH32 state from debug state forces pc bit0 to 0.

Type: `bool`

Default value: `false`

### **`align_pc_on_illegal_exception_return_to_aarch32`**

Align PC when performing an illegal exception return from AArch64 to AArch32.

Type: `bool`

Default value: `true`

### **`allow_s1_dbm_update_on_s2_mmu_fault`**

Whether s1 dirty bit update is done when s2 of ipa (not s1 ttw) generates mmu fault.



Type: `bool`

Default value: `true`

### **`amair_reg_rw_mask`**

RW mask for implementation-defined registers.

Type: `uint64_t`

Default value: `0x0`

### **`amu_aux_counter_mask`**

If ARMv8.4 is implemented, each bit of the field, 0 to 15, when 1 indicates that the corresponding virtual offset register, AMEVCNTR1<n>\_ELO/AMEVTYPER1<n>\_ELO, is implemented.

Type: `uint16_t`

Default value: `0x0`

### **`amu_aux_type_fixed`**

Lists which AMU auxiliary registers that are fixed and to which event type. The JSON schema is: {fixed\_aux\_reg:evt\_type, ...}. For example {"0":0x300} would make auxiliary register 0 fixed to event type 0x300.

Type: `string`

Default value: `N/A`

### **`amu_aux_voffset_mask`**

If ARMv8.6 is implemented, each bit of the field, 0 to 15, when 1 indicates that the corresponding virtual offset register, AMEVCNTVOFF1<n>\_EL2, is implemented.

Type: `uint16_t`

Default value: `0x0`

### **`amu_has_external_interface`**

Implement external memory-mapped access to system register of activity monitor unit from ARMv8.4 (FEAT\_AMU\_EXT).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: `0`

### **`amu_has_sysreg_interface`**

Implement system register access to activity monitor unit from ARMv8.4.

Type: `bool`

Default value: `true`

### **`amu_mmap_address`**

AMU base address for each core on system bus. 0 means the AMU is not mapped, otherwise the address must be 4KB aligned. JSON schema for the parameter value is: `{"format": "all_addrs_are_absolute_wrt_systembus", "cores": [{"amu": 0x0}, {"amu": 0x0}, {"amu": 0x0}, {"amu": 0x0}]}`.

Type: `string`

Default value: `N/A`

### **`amu_num_auxiliary_counters`**

Number of AMU auxiliary counters implemented.

Type: `uint8_t`

Default value: `0`

### **`amu_reset_domain`**

Reset domain for activity monitor unit. 0, COLD\_RESET. 1, WARM\_RESET. 2, NONE.

Type: `uint8_t`

Default value: `0`

### **`amu_version`**

Selects the activity monitor version implemented - 1, AMUv1 for Armv8.4 is implemented.- 2, AMUv1 for Armv8.6 is implemented (FEAT\_AMUv1p1).

Type: `uint8_t`

Default value: `1`

### **`apshr_read_restrict`**

At EL0, unknown bits of APSR are **RAZ**.

Type: `bool`

Default value: `false`

### **`arm_v8_7_accelerator_support_level`**

Implements accelerator support instructions: 0, Not implemented 1, FEAT\_LS64 implemented 2, FEAT\_LS64\_V implemented 3, FEAT\_LS64\_ACCDATA implemented 4, FEAT\_LS64WB implemented.

Type: `uint8_t`

Default value: 0

### **`atomic_memtype_fault_prio_less_than_gpc_fault`**

Bitmask controlling whether unsupported memtype faults are lower priority than GPC faults: bit0 applies to atomic/exclusive accesses, bit1 applies to LS64.

Type: `uint8_t`

Default value: 0

### **`atomic_memtype_fault_priority`**

This parameter describes the priority of unsupported atomic/exclusive memtype fault w.r.t alignment and permission fault. 0, BEFORE\_ALIGN\_MEM\_FAULT. 1, AFTER\_ALIGN\_BEFORE\_PERM\_FAULT. 2, AFTER\_PERM\_FAULT.

Type: `uint8_t`

Default value: 0

### **`auxilliary_feature_register0`**

Value of AFR0 ID register.

Type: `uint32_t`

Default value: 0x0

### **`branch-predictor-clear-policy`**

Set branch prediction policy as defined for MMFR1[31:28]. This does not change the behaviour of the branch predictor, only what is reported in MMFR1.BPred.

Type: `uint8_t`

Default value: 2

### **`branch-predictor-supported-ops`**

Set branch prediction policy as defined for MMFR3[11:8]. This does not change the behaviour of the branch predictor, only what is reported in MMFR3.BPMaint.

Type: `uint8_t`

Default value: 1

### **`brbe_disable_recording`**

If BRBE is implemented and this is set to true, disable BRBE recording. All registers will be functional, but no branches will be recorded. This will improve model performance for workloads that enable BRBE, but don't care about the information stored in it. (FEAT\_BRBE).

Type: `bool`

Default value: `false`

### **`brbe_log2_num_records`**

Log2 of number of BRB records supported. 3 -> 8 records, ... 6 -> 64 records.

Type: `uint8_t`

Default value: 6

### **`brbinf_type_override_on_impdef_trap_to_el3`**

If true, force BRBINF.TYPE=0x23 (trap) when ESR.EC=0x1f (implementation defined exception to EL3) (FEAT\_BRBE).

Type: `bool`

Default value: `false`

### **`bti_support_level`**

Support branch target identification: 0 - None. 1 - Support FEAT\_BTI (mandatory from ARMv8.5). 2 - Support FEAT\_BTIE.

Type: `uint8_t`

Default value: 0

### **`cache-log2linelen`**

Log2 of the cache line length in bytes.

Type: `uint8_t`

Default value: 6

### **`cache_maintenance_hits_watchpoints`**

DCIMVA operations executed in AArch32 modes hit watchpoints.

Type: `bool`

Default value: `false`

### **`cache_mlb_with_default_id`**

If true, incomplete mlb fetches that ends up with entries with default MPAM ids will still be cached.

Type: `bool`

Default value: `false`

**changing\_block\_size\_without\_bbm\_support**

Changing block size without break-before-make support level(OPTIONAL from Armv8.3): 0, Unsupported, 1, Level 1 support (FEAT\_BBML1), 2, Level 2 support (FEAT\_BBML2), 3, Level 3 support (FEAT\_BBML3).

Type: `uint8_t`

Default value: 0

**check\_memory\_attributes**

Detect and report TLB use of conflicting memory attributes for views of the same physical address.

Type: `bool`

Default value: `false`

**checked\_pointer\_arithmetic\_support\_level**

Specify the Checked Pointer Arithmetic support level: 0, not implemented. 1, FEAT\_CPA is implemented. 2, FEAT\_CPA2 is implemented.

Type: `uint8_t`

Default value: 0

**clean\_invalidate\_cache\_on\_warm\_reset**

Clean and invalidate caches on warm reset.

Type: `bool`

Default value: `false`

**clear\_IT\_when\_IL\_set**

Clear IT bits when performing a *legal* exception return to AArch32 when IL is set.

Type: `bool`

Default value: `false`

**clear\_IT\_when\_IL\_set\_explicitly**

Apart from `clear_IT_when_IL_set`, also clear IT bits when loading CPSR from SPSR/memory and IL == 1 in the value being loaded.

Type: `bool`

Default value: `false`

**clear\_ec\_in\_debug\_state**

When ARMv8.8 debug extension is implemented, whether EDESR.EC bit is set/cleared on entering debug state due to pending exception catch caused by EDESR.EC=1.

Type: `bool`

Default value: `false`

**clear\_reg\_top\_eret**

Behaviour of the upper 32-bits of the Xn registers when changing between AArch32 state and AArch64 state. 0, upper 32-bits preserved for all registers. 1, upper 32-bits set to 0 for all accessible registers. 2, upper 32-bits set to 0 for a random selection of accessible registers. 3, upper-32-bits set to 0 for registers touched in AArch32.

Type: `uint8_t`

Default value: 1

**clear\_reg\_top\_set**

Whether to clear upper 32-bits of the Xn register when corresponding AArch32 register is set via Iris.

Type: `bool`

Default value: `true`

**configure\_pmu\_events\_with\_json**

"Configure v8.6 and newer PMU events. Note : This param has high priority and overrides the setting of "has\_\*\_pmu\_events" if both the params are provided. JSON schema for the parameter value is e.g. 1. {"all":false} 2. {"pmu\_events":["EVENT\_NAME\_1","EVENT\_NAME\_2"]}".

Type: `string`

Default value: `N/A`

**configure\_v8\_6\_pmu\_events\_with\_json**

"[DEPRECATED: Set configure\_pmu\_events\_with\_json to the same value instead] Configure v8.6 PMU events. Note : This param has high priority and overrides the setting of "has\_v8\_6\_pmu\_events" if both the params are provided. JSON schema for the parameter value is e.g. 1. {"all":false} 2. {"pmu\_events":["BR\_INDNR\_RETIRE", "BR\_IND\_RETIRE", "BR\_RETURN\_SKIP\_RETIRE", "BR\_RETURN\_ANY\_RETIRE", "BR\_INDNR\_SKIP\_RETIRE", "BR\_INDNR\_TAKEN\_RETIRE", "BR\_IND\_SKIP\_RETIRE", "BR\_IND\_TAKEN\_RETIRE", "BR\_IMMED\_SKIP\_RETIRE", "BR\_IMMED\_TAKEN\_RETIRE", "BR\_SKIP\_RETIRE"]}".

Type: `string`

Default value: `N/A`

**configure\_v8\_8\_pmu\_events\_with\_json**

"[DEPRECATED: Set configure\_pmu\_events\_with\_json to the same value instead] Configure v8.8 PMU events. Note : This param has high priority and overrides the setting of "has\_v8\_8\_pmu\_events" if both the params are provided. JSON schema for the parameter value is e.g. 1. {"all":false} 2. {"pmu\_events":["BR\_HINT\_COND\_RETIRE", "BR\_COND\_TAKEN\_RETIRE", "BR\_UNCOND\_RETIRE", "BR\_COND\_RETIRE", "BRNL\_TAKEN\_RETIRE", "BRNL\_IND\_TAKEN\_RETIRE", "BRNL\_INDNR\_TAKEN\_RETIRE", "BRNL\_IMMED\_TAKEN\_RETIRE", "BL\_TAKEN\_RETIRE", "BL\_IND\_TAKEN\_RETIRE", "BL\_IMMED\_TAKEN\_RETIRE"]}]".

Type: string

Default value: N/A

**configure\_v8\_9\_pmu\_events\_with\_json**

"[DEPRECATED: Set configure\_pmu\_events\_with\_json to the same value instead] Configure v8.9 PMU events. Note : This param has high priority and overrides the setting of "has\_v8\_9\_pmu\_events" if both the params are provided. JSON schema for the parameter value is e.g. 1. {"all":false} 2. {"pmu\_events":["ASE\_SVE\_RETIRE", "ASE\_RETIRE", "VFP\_RETIRE", "SVE\_RETIRE", "CRYPTO\_RETIRE", "SIMD\_INST\_RETIRE", "ASE\_INST\_RETIRE", "SVE\_INST\_RETIRE", "ASE\_SVE\_INST\_RETIRE", "LD\_ANY\_RETIRE", "ST\_ANY\_RETIRE", "LDST\_ANY\_RETIRE", "DP\_RETIRE"]}]".

Type: string

Default value: N/A

**core\_cache\_protection**

core\_cache\_protection can change ERROFR, ERROPFGF and ERROPFGCTL fields. Possible values are: -1:Not implemented (by default), 0:Disabled, 1:Enabled.

Type: int8\_t

Default value: -1

**cpacr\_trcdis\_behaviour**

Behaviour of CPACR.TRCDIS/NSACR.NSTRCDIS when there is no CP14 ETM interface. 0, **RAZ/WI**. 2, implemented.

Type: uint8\_t

Default value: 2

**cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: uint32\_t

Default value: 0x1

### **cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: uint32\_t

Default value: 0x1

### **cpuselr\_el3\_sync\_immediate**

Adjust when the patching selection register synchronises - either immediately (true - default), or awaiting for barrier event.

Type: bool

Default value: true

### **cpy\_mops\_option**

Set option for Armv8.8 CPY(FEAT\_MOPS). 0, use default(i.e. use value configured through has\_mops\_option). 1, implemented using Option A. 2, implemented using Option B.

Type: uint8\_t

Default value: 0

### **cpyf\_mops\_option**

Set option for Armv8.8 CPYF(FEAT\_MOPS). 0, use default(i.e. use value configured through has\_mops\_option). 1, implemented using Option A. 2, implemented using Option B.

Type: uint8\_t

Default value: 0

### **cycle\_counter\_freeze\_on\_spe\_event**

If true, pmu cycle counter does not count when pmcr\_el0.dp=1 and pmu event counters are frozen by pmcr\_el0.fzs. (FEAT\_SPE\_DPFZS).

Type: bool

Default value: false

### **d128\_disabled\_ps\_resvd\_size**

Physical size treated when TCR.(I)PS is programmed with value seven and D128 is disabled via TCR. 0, 48 bits. 1, 52 bits. 2, 56.

Type: uint8\_t

Default value: 2



**data\_abort\_on\_gcs\_access\_to\_non\_normal\_memory**

If true, a GCS data access to non-normal memory results in a data abort for unsupported access.

Type: `bool`

Default value: `false`

**dbg-bcr-reserved-behavior**

This is the behavior of the reserved values of the BT field in DBGBCR. Possible values are: - 0 = Disabled. - 1 = BT[2] is ignored..

Type: `uint8_t`

Default value: 1

**dbg\_rom\_dap\_addr**

Debug ROM dap base address.

Type: `uint64_t`

Default value: 0x0

**dbgitr\_buffer\_size**

Number of instructions which can be buffered before EDSCR.ITE is cleared.

Type: `uint32_t`

Default value: 0x0

**dbgxvr\_ress\_is\_stateful**

Whether DBGWVR/DBGBVR.RESS returns last written value. if set to false, RESS returns sign extended value.

Type: `bool`

Default value: `false`

**dc\_fault\_unaligned\_s1\_device\_s2\_fwb**

Whether takes an Alignment Fault caused by the memory type on a DC {ZVA,GZVA,GVA} if the stage 1 memory type is any Device memory type.

Type: `bool`

Default value: `false`

**dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-maintenance\_latency**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-miss\_latency**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-prefetch\_enabled**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

Type: `bool`

Default value: `false`

**dcache-read\_access\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-read\_bus\_width\_in\_bytes**

L1 D-Cache read bus width in bytes used to calculate per-access timing annotations.

Type: `uint32_t`

Default value: `0x8`

**dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-size**

L1 D-Cache size in bytes.

Type: `uint32_t`

Default value: `0x8000`

**dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-state\_modelled**

Set whether D-cache has stateful implementation.

Type: `bool`

Default value: `false`

**dcache-ways**

L1 D-Cache number of ways (sets are implicit from size).

Type: `uint8_t`

Default value: `2`

**dcache-write\_access\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-write\_bus\_width\_in\_bytes**

L1 D-Cache write bus width in bytes used to calculate per-access timing annotations.

Type: `uint32_t`

Default value: `0x8`

**dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcimva\_requires\_s2\_write\_permissions**

Data-cache invalidate by MVA operations require stage 2 write permission (virtualised AArch32 guest).

Type: `bool`

Default value: `false`

**dczva\_reports\_lowest\_addr\_on\_tag\_check\_fail**

Whether DC ZVA reports lowest address in FAR on tag check fail.

Type: `bool`

Default value: `false`

**dczva\_wp\_far\_behaviour**

Set option for address stored in FAR/EDWARD after watchpoints hit by DC ZVA. - FAR recorded matches lowest watchpointed address accessed by the instruction - FAR recorded matches lowest address accessed by the instruction within same translation granule as watchpointed address - FAR recorded matches highest address accessed by the instruction within same translation granule as watchpointed address.

Type: `uint8_t`

Default value: `0`

**debug\_auth\_signals\_sampled\_at\_reset**

Debug authentication signals can be configured as either sampled at reset only or at any time for External Root Debug.

Type: `bool`

Default value: `false`

### **debug\_components\_dap\_address**

Debug components ROM,ED,CTI,PMU,TRACE and TRBU base address for each core on debug bus. The "rom" field in the "cores" array are only allowed when 'debug\_rom\_is\_flat' is false. JSON schema for the parameter value is: {"format":"all\_addrs\_are\_absolute\_wrt\_debugbus", "cores": [{"rom":0x0, "ed":0x0, "cti":0x0, "pmu":0x0, "etm":0x0, "trbu":0x0}, {"rom":0x0, "ed":0x0, "cti":0x0, "pmu":0x0, "etm":0x0, "trbu":0x0}, {"rom":0x0, "ed":0x0, "cti":0x0, "pmu":0x0, "etm":0x0, "trbu":0x0}, {"rom":0x0, "ed":0x0, "cti":0x0, "pmu":0x0, "etm":0x0, "trbu":0x0}]}

Type: `string`

Default value: `N/A`

### **debug\_components\_mmap\_address**

Debug components ROM,ED,CTI,PMU,TRACE and TRBU base address for each core on system bus. The "rom" field in the "cores" array are only allowed when 'debug\_rom\_is\_flat' is false. JSON schema for the parameter value is: {"format":"all\_addrs\_are\_absolute\_wrt\_systembus", "cores": [{"rom":0x0, "ed":0x0, "cti":0x0, "pmu":0x0, "etm":0x0, "trbu":0x0}, {"rom":0x0, "ed":0x0, "cti":0x0, "pmu":0x0, "etm":0x0, "trbu":0x0}, {"rom":0x0, "ed":0x0, "cti":0x0, "pmu":0x0, "etm":0x0, "trbu":0x0}, {"rom":0x0, "ed":0x0, "cti":0x0, "pmu":0x0, "etm":0x0, "trbu":0x0}]}

Type: `string`

Default value: `N/A`

### **debug\_entry\_is\_context\_sync**

If true, Entry in debug state is Context sync. Exiting debug state is a context synchronising operation, but entering is not. However some cpu implementation can consider also the Entry in Debug state as a CSE.

Type: `bool`

Default value: `false`

### **debug\_rom\_is\_class\_9**

If true, present a debug ROM table as a class 9 device. Otherwise, use a class 1 ROM table.

Type: `bool`

Default value: `false`

### **debug\_rom\_is\_flat**

If true, present a debug ROM table recommended by ARMv8 Debug Architecture. Otherwise, use nested ROM tables.

Type: `bool`

Default value: `false`

### **delay\_serror**

Add a propagation delay of serror signal into the core.

Type: `uint32_t`

Default value: `0x0`

### **delayed\_dbgreg\_between\_secure\_views**

If `delayed_dbgreg` is enabled, whether the secure and nonsecure external views require explicit synchronization. values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **delayed\_pmureg\_between\_secure\_views**

If `delayed_pmureg` is enabled, whether the secure and nonsecure external views require explicit synchronization. values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **dfr1\_reads\_actual\_bp\_wp\_ctx\_cmp**

If true, the register `ID_AA64DFR1_EL1/EDDFR1` reports the actual number of BRPs, WRPs and CTX\_CMPs even when the number of bp/wp/ctx\_cmp is less than 16.

Type: `bool`

Default value: `false`

### **dic-spi\_count**

Number of shared peripheral interrupts implemented.

Type: `uint8_t`

Default value: 64

### **disable\_bti\_effects**

When BTI effects are disabled: `PSTATE.BTYPE` is never updated and BTI exceptions are never triggered. This is not architecturally correct behaviour, but can lead to improved simulation performance.

Type: `bool`

Default value: `false`

### **`disable_sve_plugin`**

If true, SVE will not be implemented in this processor even if the plugin is loaded (FEAT\_SVE).

Type: `bool`

Default value: `false`

### **`disable_unknown_update_event_on_reset`**

Disables SYSREG\_UPDATE event notification on reset for the registers whose bitfields are all reserved or resets to architecturally unknown value.

Type: `bool`

Default value: `false`

### **`dsb_accumulate_threshold`**

Limit the maximum number of observable DSB side effects which can be queued (e.g. TLBI), after which DSB sync will be done automatically.

Type: `uint32_t`

Default value: `0x100`

### **`e2h_forces_interrupt_overrides`**

If true, HCR\_EL2.xMO are treated as 1 else as programmed.

Type: `bool`

Default value: `false`

### **`early_implicit_error_sync_event_behaviour`**

Set option for Early Implicit Error Synchronization event (FEAT\_IESB) 0x0 - Behavior is not described ID\_AA64MMFR4\_EL1.EIESB = 0x00x1 - Implicit Error synchronization event is inserted before an exception is taken to EL3 (depending on SCR\_EL3.NMEA) ID\_AA64MMFR4\_EL1.EIESB = 0x10x2 - Implicit Error synchronization event is inserted before an exception is taken to ELx (depending on SCR\_EL3.NMEA and SCTLR2\_ELx.NMEA) ID\_AA64MMFR4\_EL1.EIESB = 0x20xF - Implicit Error synchronization event is inserted after an exception is taken ID\_AA64MMFR4\_EL1.EIESB = 0xF.

Type: `uint8_t`

Default value: `0`

### **`ecv_support_level`**

Implement Enhanced Counter Virtualization feature from ARMv8.6. 0, Not supported. 1, fully supported without CNTPOFF. 2, fully supported with CNTPOFF (FEAT\_ECV).

Type: `uint8_t`

Default value: 0

### **`eddfr1_reads_idreg_mask`**

Mask to configure each bitfield for EDDFR1 register, whether to be read from corresponding bitfield in AA64DFR1 register. Note: Only those bitfield which supports 'same as ID register or res0' functionality can be configured.

Type: `uint64_t`

Default value: `0xffffffffffffffff`

### **`eddfr2_reads_idreg_mask`**

Mask to configure each bitfield for EDDFR2 register, whether to be read from corresponding bitfield in ID\_AA64DFR2 register. Note: Only those bitfield which supports 'same as ID register or res0' functionality can be configured.

Type: `uint64_t`

Default value: `0xffffffffffffffff`

### **`eddfr_reads_idreg_mask`**

Mask to configure each bitfield for EDDFR register, whether to be read from corresponding bitfield in AA64DFR register. Note: Only those bitfield which supports 'same as ID register or res0' functionality can be configured.

Type: `uint64_t`

Default value: `0xffffffffffffffff`

### **`edpfr_ras_unknown_bits_read_as_0`**

If true then **UNKNOWN** bits in RAS field in EDPFR are read as 0.

Type: `bool`

Default value: `false`

### **`edxfr_reads_idreg`**

Whether EDDFR, EDDFR1 and EDDFR2 reads corresponding bitfield value from ID\_AA64DFR reg. Also, when this parameter is enabled, bitfields of these registers are configurable through 'eddfr\*\_reads\_idreg\_mask' parameter.

Type: `bool`

Default value: `false`



**e10\_can\_access\_imp\_def\_functionality**

If not made UNDEF by `imp_def_functionality_behaviour`, EL0 can access **IMPLEMENTATION DEFINED** registers and system instructions.

Type: `bool`

Default value: `false`

**e10\_el1\_only\_non\_secure**

Secure/non-secure state if EL2 and EL3 are not implemented. 0, secure. 1, non-secure.

Type: `bool`

Default value: `false`

**e13\_trap\_priority\_when\_secure\_debug\_disabled**

Undef when secure debug is disabled (`EDSCR.SDD == 1`) && boolean `IMPLEMENTATION_DEFINED` 'EL3 trap priority when `SDD == 1`'.

Type: `bool`

Default value: `false`

**enable-gicv5**

if `enable-gicv5` is set, then GICv5 is Supported.

Type: `bool`

Default value: `false`

**enable\_address\_contig\_check**

Check the input address range for the table entries that have the contiguous hint bit set.

Type: `bool`

Default value: `false`

**enable\_debug\_access\_trace**

If true, enables traces on debug access. Currently enables following traces on debug :: `MMU_TRANS`.

Type: `bool`

Default value: `false`

**enable\_debug\_auth\_signals\_config**

Debug Authentication Signals DBGEN, SPIDEN (and if RME is enabled RLPIDEN and RTPIDEN) are configurable (default) or not configurable, (hardwired to 1). This parameter is the integer representation of a bitmap to enable configuration of these signals, with:- BIT[0] = DBGEN- BIT[1] = SPIDEN- BIT[2] = RLPIDEN- BIT[3] = RTPIDEN.

Type: `uint8_t`

Default value: 15

**enable\_mpam\_mvms\_mlb\_cache**

If true, MPAMv2\_VID MLB entries for MVMS (VPARTID/VPMG \342\206\222 MITT base) are always cached until invalidation, otherwise not cached.

Type: `bool`

Default value: `true`

**enable\_mpam\_vid\_pid\_mlb\_cache**

If true, MPAMv2\_VID MLB entries for VPARTID/VPMG \342\206\222 physical PARTID/PMG are always cached until invalidation, otherwise not cached.

Type: `bool`

Default value: `true`

**enable\_tlb\_contig\_check**

Perform extra pagetable walks to check translation table entries that have the contiguous hint bit set. Ignored if FEAT\_BBML3 is implemented.

Type: `bool`

Default value: `false`

**enhanced\_pac2\_level**

Implements Enhanced PAC2 from ARMv8.6 (FEAT\_PAuth2), and PAC enhancements from ARMv9.5 (FEAT\_PAuth\_LR). options 0-3 of this feature are mandatory for ARMv8.6 but can be cherry-picked to a ARMv8.3(or greater) implementation. FEAT\_FPACCOMBINE is mandatory in the presence of Future Architecture Technologies (FAT). 0: No EnhancedPAC2, 1: EnhancedPAC2 Only (FEAT\_PAuth2), 2: EnhancedPAC2 with FPAC (FEAT\_FPAC), 3: EnhancedPAC2 with FPACCombined (FEAT\_FPACCOMBINE), 4: EnhancedPAC2 with LR signing (FEAT\_PAuth\_LR).

Type: `uint8_t`

Default value: 0

**error\_record\_feature\_register**

RAS feature register values. An array of JSON objects. The JSON schema for the array is:  
 [{"ED":0x0, "IMPDEF\_3\_2":0x0, "UI":0x0, "FI":0x0, "UE":0x0, "CFI":0x0, "CEC":0x0, "RP":0x0, "DUI":0x0, "CEO":0x0, "CI":0x0, "TS":0x0, "INJ":0x0, "FRX":0x0, "UC":0x0, "UEU":0x0, "UER":0x0, "UEO":0x0, "DE":0x0, "CE":0x0, "Visibility":"Core"},other\_feature\_register\_values].  
 Where ED,UI,FI,CE and UE have valid values between 0x0 - 0x3. CFI and DUI have valid values 0x0, 0x2 and 0x3. CEC has valid values 0x0,0x2 or 0x4. RP,CEO,INJ,FRX,UC,UEU,UER,UEO,DE has valid values 0x0 or 0x1. CI and TS has valid values of 0x0, 0x1 and 0x2. Visibility has valid values "Core" or "Cluster".

Type: string

Default value: N/A

**error\_record\_feature\_register\_json\_file**

File path to the RAS feature register values as JSON. The file uses the same format as the error\_record\_feature\_register parameter value.

Type: string

Default value: N/A

**erxpfctl\_res0\_stateful\_mask**

Mask for stateful bits for ERXPFCTL which are **RES0**.

Type: uint64\_t

Default value: 0x0

**esr\_write\_update\_res0**

If true, and RASv2 is enabled, then ESR\_ELx.WU field is **RES0** for errors on both loads and stores (FEAT\_RASv2).

Type: bool

Default value: false

**ete.ASYNC\_PACKETS\_WHEN\_VIEWINST\_OFF**

Generate the non-periodic alignment synchronisation packet generation when trace unit is operative.

Type: bool

Default value: false

**ete.ATBTRIG**

ATB trigger support.

Type: `bool`

Default value: `true`

#### **`ete.CCITMIN`**

Minimum cycle count value.

Type: `uint16_t`

Default value: `0x4`

#### **`ete.CCSIZE`**

Cycle counter size.

Type: `uint8_t`

Default value: `12`

#### **`ete.CLAIMTAGS`**

Number of claim tags.

Type: `uint8_t`

Default value: `8`

#### **`ete.COMMOPT`**

Commit mode.

Type: `bool`

Default value: `true`

#### **`ete.DEBUG`**

DEBUG.

Type: `uint8_t`

Default value: `2`

#### **`ete.DESIGNER`**

DESIGNER value.

Type: `uint8_t`

Default value: `65`

#### **`ete.ETE_REVISION`**

ETE revision: 0=ETEv1.0, 1=ETEv1.1, 2=ETEv1.2, 3=ETEv1.3.

Type: `uint8_t`

Default value: 0

#### **`ete.EXCEPTION_WITH_CONTEXT`**

Whether `EXCEPTION_WITH_CONTEXT` packet is supported.

Type: `bool`

Default value: `true`

#### **`ete.EXPLICITLY_COMMIT_PO_ELEMS`**

Whether to unilaterally explicitly emit a commit after a PO packet.

Type: `bool`

Default value: `false`

#### **`ete.IMPDEFEXCEPPERCENTAGE`**

Percentage of `IMPDEF` exceptions inserted in instruction blocks.

Type: `uint8_t`

Default value: 0

#### **`ete.IMPDEF_TRACE_ON`**

Whether trace is flushed and trace on packet generated by events described by bitmap value. bit 0 - PE entering low power state, bit 1 - PE entering debug state.

Type: `uint8_t`

Default value: 0

#### **`ete.IMPRECISE_FILTERING`**

Number of instruction blocks traced on a transition in the filtering.

Type: `uint8_t`

Default value: 0

#### **`ete.LPOVERRIDE`**

Low power override.

Type: `bool`

Default value: `true`

**ete.MAXSPEC**

Maximum speculation depth.

Type: `uint32_t`

Default value: `0x0`

**ete.MAX\_INST\_PER\_Q**

Maximum limit for the number of instructions implied by a Q element.

Type: `uint16_t`

Default value: `0x1`

**ete.NOOVERFLOW**

No overflow.

Type: `bool`

Default value: `false`

**ete.NUMACPAIRS**

Number of instruction address comparators pairs.

Type: `uint8_t`

Default value: `4`

**ete.NUMCIDC**

Number of context ID comparators.

Type: `uint8_t`

Default value: `1`

**ete.NUMCNTR**

Number of counters.

Type: `uint8_t`

Default value: `2`

**ete.NUMEXTINSEL**

Number of external input selectors.

Type: `uint8_t`

Default value: `4`

**ete.NUMPC**

Number of PE comparators.

Type: uint8\_t

Default value: 0

**ete.NUMSEQSTATE**

Number of sequencer states.

Type: uint8\_t

Default value: 4

**ete.NUMSSCC**

Number of single shot comparators.

Type: uint8\_t

Default value: 1

**ete.NUMVMIDC**

Number of virtual ID comparators.

Type: uint8\_t

Default value: 1

**ete.NumberOfETEEEvents**

Number of trace events.

Type: uint8\_t

Default value: 2

**ete.NumberOfRSPairs**

Number of resource selector pairs.

Type: uint8\_t

Default value: 8

**ete.PIDR\_CM0D**

TRCPIDR CM0D value.

Type: uint8\_t

Default value: 0

**ete.PIDR\_DESIGNER**

TRCPIDR DESIGNER value.

Type: uint16\_t

Default value: 0x0

**ete.PIDR\_PART**

TRCPIDR PART number value.

Type: uint16\_t

Default value: 0x0

**ete.PIDR\_REVAND**

TRCPIDR REVAND value.

Type: uint8\_t

Default value: 0

**ete.PIDR\_REVISION**

TRCPIDR REVISION value.

Type: uint8\_t

Default value: 0

**ete.QFILT**

Q filtering.

Type: bool

Default value: false

**ete.QSUP**

Q support.

Type: uint8\_t

Default value: 0

**ete.Q\_CADENCE**

Number of instruction blocks traced between two Q elements.

Type: uint16\_t

Default value: 0x1



**ete.RAZWI\_REG\_SEL\_TOP\_BIT**

Implement Resource Selectors or Resource Selector Pairs bits as **RAZ/WI**.

Type: `bool`

Default value: `false`

**ete.REGS\_WRITE\_IGNORE\_WHEN\_ENABLED**

Whether direct and external writes to registers except TRCPRGCTLR, TRCCLAIMSET and TRCCLAIMCLR are ignored when not in Idle state.

Type: `bool`

Default value: `false`

**ete.REG\_ACCESS\_ONLY\_MODE**

If enabled, all traces are disabled. Plugin only allows register accesses.

Type: `bool`

Default value: `false`

**ete.RES0\_STATEFUL**

Whether **RES0** bits are stateful or **RAZ/WI**.

Type: `bool`

Default value: `false`

**ete.RETSTACK**

Return stack depth.

Type: `uint8_t`

Default value: 3

**ete.REVISION**

TRCIDR1 revision value.

Type: `uint8_t`

Default value: 0

**ete.SIM\_OVERFLOW\_GRANULARITY**

Number of instruction blocks in each granule, for simulated overflow.

Type: `uint32_t`

Default value: 0x64

**ete.SIM\_OVERFLOW\_PERCENTAGE**

Percentage of instruction blocks lost in each granule, for simulated overflow.

Type: uint8\_t

Default value: 0

**ete.SOURCE\_ADDRESS**

Allow generation of source address elements.

Type: bool

Default value: false

**ete.STALLCTRL**

Stall control.

Type: bool

Default value: true

**ete.SYSSTALL**

System stall.

Type: bool

Default value: true

**ete.TRACEIDSIZE**

Trace ID size.

Type: uint8\_t

Default value: 7

**ete.TRACE\_OUTPUT**

File to which to write trace byte stream.

Type: string

Default value: N/A

**ete.TRACE\_OUTPUT\_ENABLE**

ETE Trace output enable: 1=enable, 0=disable.

Type: bool

Default value: `false`

**ete.TRCRSRTA\_FORCED\_EXCEP**

TRCRSR.TA value for a forcibly traced exception.

Type: `bool`

Default value: `false`

**ete.TSMARK**

Whether timestamp markers are supported.

Type: `bool`

Default value: `false`

**ete.TSSIZE**

Timestamp size.

Type: `uint8_t`

Default value: `8`

**ete.WFXMODE**

WFX mode.

Type: `bool`

Default value: `true`

**ets\_level**

Level of Enhanced Translation Synchronization support. 0: FEAT\_ETS2 not supported, 1: FEAT\_ETS2 not supported, 2: FEAT\_ETS2 supported, 3: FEAT\_ETS3 supported.

Type: `uint8_t`

Default value: `0`

**exception\_catch\_before\_software\_step**

Exception catch priority for the exception trapping form of exception catch (Armv8.2 or later, or `exception_catch_type=0`). If true, the exception catch debug event has higher priority than software step and halting step.

Type: `bool`

Default value: `true`

**exception\_catch\_type**

Type of exception catch (ARMv8.0 - ARMv8.1 only). 0, exception trapping. 1, non-exception trapping, higher priority than step. 2, non-exception trapping, lower priority than step.

Type: `uint8_t`

Default value: 0

**exclusive\_monitor\_clear\_on\_atomic\_from\_same\_master**

Exclusive monitors in the cluster will be cleared by a atomic by the same master to the monitored address.

Type: `bool`

Default value: `true`

**exclusive\_monitor\_clear\_on\_store\_from\_same\_master**

Exclusive monitors in the cluster will be cleared by a store by the same master to the monitored address.

Type: `bool`

Default value: `true`

**exclusive\_monitor\_clear\_on\_strex\_address\_mismatch**

Exclusive monitors in the cluster will be cleared when a strex fails because the address does not match.

Type: `bool`

Default value: `true`

**exclusive\_monitor\_clear\_on\_strex\_success**

Exclusive monitors in the cluster will be cleared when a strex succeeds.

Type: `bool`

Default value: `true`

**exercise\_stxr\_fail**

Controls the rejection of exclusive store instructions. 0: exclusive store instructions should behave as normal, 1: Reject a pseudo-random majority of exclusive store instructions, 2: Always fail exclusive store instructions.

Type: `uint8_t`

Default value: 0

**ext\_abort\_device\_GRE\_prefetch\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as ext\_abort\_prefetch\_ras\_index, Valid indices in range [0, number\_of\_error\_records-1].

Type: `int16_t`

Default value: -1

**ext\_abort\_device\_GRE\_prefetch\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as ext\_abort\_prefetch\_ras\_type, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `int8_t`

Default value: -1

**ext\_abort\_device\_GRE\_read\_is\_critical**

Critical reporting of device-GRE read external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_device\_GRE\_read\_is\_sync**

Synchronous reporting of device-GRE read external aborts. 0, asynchronous. 1, synchronous. 2, same as ext\_abort\_device\_read\_is\_sync.

Type: `uint8_t`

Default value: 2

**ext\_abort\_device\_GRE\_read\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as ext\_abort\_device\_read\_ras\_index, Valid indices in range [0, number\_of\_error\_records-1].

Type: `int16_t`

Default value: -1

**ext\_abort\_device\_GRE\_read\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as ext\_abort\_device\_read\_ras\_type, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `int8_t`

Default value: -1

**ext\_abort\_device\_GRE\_write\_is\_critical**

Critical reporting of device-GRE write external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_device\_GRE\_write\_is\_sync**

Synchronous reporting of device-GRE write external aborts. 0, asynchronous. 1, synchronous. 2, same as `ext_abort_device_write_is_sync`.

Type: `uint8_t`

Default value: 2

**ext\_abort\_device\_GRE\_write\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as `ext_abort_device_write_ras_index`, Valid indices in range [0, `number_of_error_records`-1].

Type: `int16_t`

Default value: -1

**ext\_abort\_device\_GRE\_write\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as `ext_abort_device_write_ras_type`, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `int8_t`

Default value: -1

**ext\_abort\_device\_nGRE\_prefetch\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as `ext_abort_prefetch_ras_index`, Valid indices in range [0, `number_of_error_records`-1].

Type: `int16_t`

Default value: -1

**ext\_abort\_device\_nGRE\_prefetch\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as `ext_abort_prefetch_ras_type`, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `int8_t`

Default value: -1

**ext\_abort\_device\_nGRE\_read\_is\_critical**

Critical reporting of device-nGRE read external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_device\_nGRE\_read\_is\_sync**

Synchronous reporting of device-nGRE read external aborts. 0, asynchronous. 1, synchronous. 2, same as `ext_abort_device_read_is_sync`.

Type: `uint8_t`

Default value: 2

**ext\_abort\_device\_nGRE\_read\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as `ext_abort_device_read_ras_index`, Valid indices in range [0, `number_of_error_records`-1].

Type: `int16_t`

Default value: -1

**ext\_abort\_device\_nGRE\_read\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as `ext_abort_device_read_ras_type`, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `int8_t`

Default value: -1

**ext\_abort\_device\_nGRE\_write\_is\_critical**

Critical reporting of device-nGRE write external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_device\_nGRE\_write\_is\_sync**

Synchronous reporting of device-nGRE write external aborts. 0, asynchronous. 1, synchronous. 2, same as `ext_abort_device_write_is_sync`.

Type: `uint8_t`

Default value: 2

**ext\_abort\_device\_nGRE\_write\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as ext\_abort\_device\_write\_ras\_index, Valid indices in range [0, number\_of\_error\_records-1].

Type: `int16_t`

Default value: -1

**ext\_abort\_device\_nGRE\_write\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as ext\_abort\_device\_write\_ras\_type, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `int8_t`

Default value: -1

**ext\_abort\_device\_prefetch\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as ext\_abort\_prefetch\_ras\_index, Valid indices in range [0, number\_of\_error\_records-1].

Type: `int16_t`

Default value: -1

**ext\_abort\_device\_prefetch\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as ext\_abort\_prefetch\_ras\_type, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `int8_t`

Default value: -1

**ext\_abort\_device\_read\_acquire\_is\_sync**

Synchronous reporting of device read with acquire external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_device\_read\_is\_critical**

Critical reporting of device-nGnRE read external aborts.

Type: `bool`

Default value: `false`



**ext\_abort\_device\_read\_is\_sync**

Synchronous reporting of device-nGnRE read external aborts.

Type: `bool`

Default value: `true`

**ext\_abort\_device\_read\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number\_of\_error\_records-1].

Type: `uint16_t`

Default value: `0x0`

**ext\_abort\_device\_read\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `uint8_t`

Default value: `0`

**ext\_abort\_device\_write\_is\_critical**

Critical reporting of device-nGnRE write external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_device\_write\_is\_sync**

Synchronous reporting of device-nGnRE write external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_device\_write\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number\_of\_error\_records-1].

Type: `uint16_t`

Default value: `0x0`

**ext\_abort\_device\_write\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO , 4 = UER, 5 = CE.

Type: `uint8_t`

Default value: 0

**ext\_abort\_fill\_data**

Returned data, if external aborts are asynchronous.

Type: `uint64_t`

Default value: 0xfdfdfdfdfcfdfdfd

**ext\_abort\_in\_virtual\_tag\_reads\_is\_sync**

Behaviour of external aborts generated by virtual tag reads. 0, asynchronous abort. 1, synchronous abort. 2, same as `ext_abort_normal_cacheable_read_is_sync`.

Type: `uint8_t`

Default value: 2

**ext\_abort\_in\_virtual\_tag\_writes\_is\_sync**

Behaviour of external aborts generated by virtual tag writes. 0, asynchronous abort. 1, synchronous abort. 2, same as `ext_abort_normal_cacheable_write_is_sync`.

Type: `uint8_t`

Default value: 2

**ext\_abort\_normal\_cacheable\_read\_is\_critical**

Critical reporting of normal write-back cacheable-read external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_normal\_cacheable\_read\_is\_sync**

Synchronous reporting of normal write-back cacheable-read external aborts.

Type: `bool`

Default value: `true`

**ext\_abort\_normal\_cacheable\_read\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number\_of\_error\_records-1].

Type: `uint16_t`

Default value: `0x0`

**ext\_abort\_normal\_cacheable\_read\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `uint8_t`

Default value: `0`

**ext\_abort\_normal\_cacheable\_write\_is\_critical**

Critical reporting of normal write-back cacheable write external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_normal\_cacheable\_write\_is\_sync**

Synchronous reporting of normal write-back cacheable write external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_normal\_cacheable\_write\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number\_of\_error\_records-1].

Type: `uint16_t`

Default value: `0x0`

**ext\_abort\_normal\_cacheable\_write\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `uint8_t`

Default value: `0`

**ext\_abort\_normal\_noncacheable\_prefetch\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as ext\_abort\_prefetch\_ras\_index, Valid indices in range [0, number\_of\_error\_records-1].

Type: int16\_t

Default value: -1

**ext\_abort\_normal\_noncacheable\_prefetch\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as ext\_abort\_prefetch\_ras\_type, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: int8\_t

Default value: -1

**ext\_abort\_normal\_noncacheable\_read\_is\_critical**

Critical reporting of normal noncacheable-read external aborts.

Type: bool

Default value: false

**ext\_abort\_normal\_noncacheable\_read\_is\_sync**

Synchronous reporting of normal noncacheable-read external aborts.

Type: bool

Default value: true

**ext\_abort\_normal\_noncacheable\_read\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number\_of\_error\_records-1].

Type: uint16\_t

Default value: 0x0

**ext\_abort\_normal\_noncacheable\_read\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: uint8\_t

Default value: 0

**ext\_abort\_normal\_noncacheable\_write\_is\_critical**

Critical reporting of normal noncacheable write external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_normal\_noncacheable\_write\_is\_sync**

Synchronous reporting of normal noncacheable write external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_normal\_noncacheable\_write\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number\_of\_error\_records-1].

Type: `uint16_t`

Default value: `0x0`

**ext\_abort\_normal\_noncacheable\_write\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `uint8_t`

Default value: `0`

**ext\_abort\_normal\_wt\_cacheable\_prefetch\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as `ext_abort_prefetch_ras_index`, Valid indices in range [0, number\_of\_error\_records-1].

Type: `int16_t`

Default value: `-1`

**ext\_abort\_normal\_wt\_cacheable\_prefetch\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as `ext_abort_prefetch_ras_type`, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `int8_t`

Default value: `-1`

**ext\_abort\_normal\_wt\_cacheable\_read\_is\_critical**

Critical reporting of normal write-through cacheable-read external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_normal\_wt\_cacheable\_read\_is\_sync**

Synchronous reporting of normal write-through read external aborts. 0, asynchronous. 1, synchronous. 2, same as `ext_abort_normal_cacheable_read_is_sync`.

Type: `uint8_t`

Default value: 2

**ext\_abort\_normal\_wt\_cacheable\_read\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as `ext_abort_normal_cacheable_read_ras_index`, Valid indices in range [0, `number_of_error_records`-1].

Type: `int16_t`

Default value: -1

**ext\_abort\_normal\_wt\_cacheable\_read\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as `ext_abort_normal_cacheable_read_ras_type`, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `int8_t`

Default value: -1

**ext\_abort\_normal\_wt\_cacheable\_write\_is\_critical**

Critical reporting of normal write-through write external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_normal\_wt\_cacheable\_write\_is\_sync**

Synchronous reporting of normal write-through write external aborts. 0, asynchronous. 1, synchronous. 2, same as `ext_abort_normal_cacheable_write_is_sync`.

Type: `uint8_t`

Default value: 2

**ext\_abort\_normal\_wt\_cacheable\_write\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as ext\_abort\_normal\_cacheable\_write\_ras\_index, Valid indices in range [0, number\_of\_error\_records-1].

Type: int16\_t

Default value: -1

**ext\_abort\_normal\_wt\_cacheable\_write\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as ext\_abort\_normal\_cacheable\_write\_ras\_type, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: int8\_t

Default value: -1

**ext\_abort\_prefetch\_device\_GRE\_read\_is\_critical**

Critical reporting of external aborts generated by device-GRE instruction fetches.

Type: bool

Default value: false

**ext\_abort\_prefetch\_device\_GRE\_read\_is\_sync**

Behaviour of external aborts generated by device-GRE instruction fetches. 0, asynchronous abort. 1, synchronous abort. 2, same as ext\_abort\_prefetch\_is\_sync.

Type: uint8\_t

Default value: 2

**ext\_abort\_prefetch\_device\_nGRE\_read\_is\_critical**

Critical reporting of external aborts generated by device-nGRE instruction fetches.

Type: bool

Default value: false

**ext\_abort\_prefetch\_device\_nGRE\_read\_is\_sync**

Behaviour of external aborts generated by device-nGRE instruction fetches. 0, asynchronous abort. 1, synchronous abort. 2, same as ext\_abort\_prefetch\_is\_sync.

Type: uint8\_t

Default value: 2

**ext\_abort\_prefetch\_device\_read\_is\_critical**

Critical reporting of external aborts generated by device-nGnRE instruction fetches.

Type: `bool`

Default value: `false`

**ext\_abort\_prefetch\_device\_read\_is\_sync**

Behaviour of external aborts generated by device-nGnRE instruction fetches. 0, asynchronous abort. 1, synchronous abort. 2, same as `ext_abort_prefetch_is_sync`.

Type: `uint8_t`

Default value: 2

**ext\_abort\_prefetch\_is\_critical**

Critical reporting of external aborts generated by normal writeback cacheable instruction fetches.

Type: `bool`

Default value: `false`

**ext\_abort\_prefetch\_is\_sync**

Behaviour of external aborts generated by normal writeback cacheable instruction fetches. 0, asynchronous abort. 1, synchronous abort.

Type: `bool`

Default value: `true`

**ext\_abort\_prefetch\_noncacheable\_read\_is\_critical**

Critical reporting of external aborts generated by normal noncacheable instruction fetches.

Type: `bool`

Default value: `false`

**ext\_abort\_prefetch\_noncacheable\_read\_is\_sync**

Behaviour of external aborts generated by normal noncacheable instruction fetches. 0, asynchronous abort. 1, synchronous abort. 2, same as `ext_abort_prefetch_is_sync`.

Type: `uint8_t`

Default value: 2



**ext\_abort\_prefetch\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number\_of\_error\_records-1].

Type: `uint16_t`

Default value: `0x0`

**ext\_abort\_prefetch\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `uint8_t`

Default value: `0`

**ext\_abort\_prefetch\_so\_read\_is\_critical**

Critical reporting of external aborts generated by device-nGnRnE instruction fetches.

Type: `bool`

Default value: `false`

**ext\_abort\_prefetch\_so\_read\_is\_sync**

Behaviour of external aborts generated by device-nGnRnE instruction fetches. 0, asynchronous abort. 1, synchronous abort. 2, same as `ext_abort_prefetch_is_sync`.

Type: `uint8_t`

Default value: `2`

**ext\_abort\_prefetch\_wt\_cacheable\_read\_is\_critical**

Critical reporting of external aborts generated by normal writethrough cacheable instruction fetches.

Type: `bool`

Default value: `false`

**ext\_abort\_prefetch\_wt\_cacheable\_read\_is\_sync**

Behaviour of external aborts generated by normal writethrough cacheable instruction fetches. 0, asynchronous abort. 1, synchronous abort. 2, same as `ext_abort_prefetch_is_sync`.

Type: `uint8_t`

Default value: `2`

**ext\_abort\_so\_prefetch\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as ext\_abort\_prefetch\_ras\_index, Valid indices in range [0, number\_of\_error\_records-1].

Type: int16\_t

Default value: -1

**ext\_abort\_so\_prefetch\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as ext\_abort\_prefetch\_ras\_type, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: int8\_t

Default value: -1

**ext\_abort\_so\_read\_is\_critical**

Critical reporting of device-nGnRnE read external aborts.

Type: bool

Default value: false

**ext\_abort\_so\_read\_is\_sync**

Synchronous reporting of device-nGnRnE read external aborts.

Type: bool

Default value: true

**ext\_abort\_so\_read\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number\_of\_error\_records-1].

Type: uint16\_t

Default value: 0x0

**ext\_abort\_so\_read\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: uint8\_t

Default value: 0

**ext\_abort\_so\_write\_is\_critical**

Critical reporting of device-nGnRnE write external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_so\_write\_is\_sync**

Synchronous reporting of device-nGnRnE write external aborts.

Type: `bool`

Default value: `true`

**ext\_abort\_so\_write\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number\_of\_error\_records-1].

Type: `uint16_t`

Default value: `0x0`

**ext\_abort\_so\_write\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `uint8_t`

Default value: `0`

**ext\_abort\_ttw\_cacheable\_read\_is\_critical**

Critical reporting of TTW cacheable read external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_ttw\_cacheable\_read\_is\_sync**

Synchronous reporting of TTW cacheable read external aborts.

Type: `bool`

Default value: `true`

**ext\_abort\_ttw\_cacheable\_read\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number\_of\_error\_records-1].

Type: `uint16_t`

Default value: `0x0`

#### **`ext_abort_ttw_cacheable_read_ras_type`**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `uint8_t`

Default value: `0`

#### **`ext_abort_ttw_noncacheable_read_is_critical`**

Critical reporting of TTW noncacheable read external aborts.

Type: `bool`

Default value: `false`

#### **`ext_abort_ttw_noncacheable_read_is_sync`**

Synchronous reporting of TTW noncacheable read external aborts.

Type: `bool`

Default value: `true`

#### **`ext_abort_ttw_noncacheable_read_ras_index`**

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number\_of\_error\_records-1].

Type: `uint16_t`

Default value: `0x0`

#### **`ext_abort_ttw_noncacheable_read_ras_type`**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `uint8_t`

Default value: `0`

#### **`ext_abort_ttw_wt_cacheable_read_is_critical`**

Critical reporting of TTW write-through cacheable read external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_ttw\_wt\_cacheable\_read\_is\_sync**

Synchronous reporting of TTW write-through cacheable read external aborts. 0, asynchronous. 1, synchronous. 2, same as ext\_abort\_ttw\_cacheable\_read\_is\_sync.

Type: `uint8_t`

Default value: 2

**ext\_abort\_ttw\_wt\_cacheable\_read\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as ext\_abort\_ttw\_cacheable\_read\_ras\_index, Valid indices in range [0, number\_of\_error\_records-1].

Type: `int16_t`

Default value: -1

**ext\_abort\_ttw\_wt\_cacheable\_read\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as ext\_abort\_ttw\_cacheable\_read\_ras\_type, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `int8_t`

Default value: -1

**external\_debug\_request\_delay**

Configure External Debug Request delay in CPU cycles.

Type: `uint32_t`

Default value: 0x0

**external\_oslar\_access\_disabled\_by\_authentication**

If true, external accesses to OSLAR, when external debugging is not enabled, will generate an error (FEAT\_Debugv8p2).

Type: `bool`

Default value: `false`

**far\_unchanged\_for\_stack\_alignment\_fault**

If true and a Stack Pointer Alignment Fault occurs, FAR\_ELx is not updated, whatever value it had is maintained.

Type: `bool`

Default value: `false`

**far\_unchanged\_for\_virtual\_serror**

If true and a virtual SError occurs, FAR\_ELx is not updated, whatever value it had is maintained.

Type: `bool`

Default value: `false`

**far\_unchanged\_when\_fnv\_set**

If true and ESR\_ELx.FnV=1, FAR\_ELx is not updated but PFAR\_ELx/MFAR\_ELx can be updated.

Type: `bool`

Default value: `false`

**fault\_on\_misprogrammed\_gpt\_contig\_region**

Whether GPF faults occur when GPT contiguous entries are misprogrammed.

Type: `bool`

Default value: `false`

**fault\_on\_nT\_bit\_set**

Whether block translation table entries with the nT bit set should always fault. Only applies when `changing_block_size_without_bbm_support_level` is 1 or higher.

Type: `bool`

Default value: `true`

**fault\_unalign\_to\_unsupported\_access**

If `has_unaligned_single_copy_atomicity` is true, whether unaligned A64 atomic, exclusive and acquire/release instructions to non iWB-oWB or the access crossing a 16-byte boundary generate fault. Bits 0,1,2,3 should be set accordingly to enable the fault behaviour. bit 0: atomic access should fault, bit 1: exclusive access should fault, bit 2: acquire/release should fault, bit 3: the access crossing a 16-byte boundary should fault.

Type: `uint8_t`

Default value: 8

**fault\_unaligned\_s1\_device\_s2\_fwb**

Whether unaligned fault with stage1 Device memory and final memory attribute forced to normal by FWB. 0, No fault. 1, will fault. 2 No fault if final Shareability is NSH.

Type: `uint8_t`

Default value: 0

**fgdt\_index\_width**

Implemented FGDT Index width value, 0: 3 bits. 1: 4 bits. 2: 5 bits.

Type: `uint8_t`

Default value: 0

**force\_align\_pc**

**UNPREDICTABLE** branch to non-word-aligned address in ARM state is forced to be aligned.

Type: `bool`

Default value: `false`

**force\_deterministic\_irg\_tag\_generation**

Force the random tag generated by the IRG instruction when GCR\_EL1.RRND=1 to equal RGSRR\_EL1.SEED[3:0] rather than a non-deterministic value.

Type: `bool`

Default value: `false`

**force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks**

Force MTE tag accesses to **RAZ/WI** and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

Type: `bool`

Default value: `false`

**force\_pstate\_pm\_in\_debug\_state**

If true, PSTATE.PM is forced to 1 while entering in debug state (FEAT\_EBEP).

Type: `bool`

Default value: `false`

**force\_sync\_on\_wfx**

If true, the PE does a context synchronization before entering low power state(WFI/WFE).

Type: `bool`

Default value: `false`

**force\_wnr\_read\_unsupported\_exclusive\_or\_atomic**

DEPRECATED, please use `force_wnr_read_unsupported_exclusive_or_atomic` instead. Whether ESR\_ELx.WnR is forced to 0 for unsupported atomic and exclusives.

Type: `bool`

Default value: `false`

**force\_wnr\_unsupported\_atomic\_hwu**

If not 0, force ESR\_ELx.WnR to a specific value on Unsupported Atomic Hardware Update. Possible values: 0: Not forced. 1: Write. 2: Read.

Type: `uint8_t`

Default value: 0

**force\_wnr\_unsupported\_exclusive\_or\_atomic**

If not 0, force ESR\_ELx.WnR to a specific value on Unsupported Atomic or Exclusives. Possible values: 0: Not forced. 1: Write. 2: Read.

Type: `uint8_t`

Default value: 0

**force\_zero\_PSTATE\_PAN**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

**force\_zero\_mpam\_partid\_and\_pmg**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

Type: `uint8_t`

Default value: 0

**fp8\_support\_level**

0->No support for Advanced SIMD, SVE2 FP8 instructions 1->Support for FEAT\_FP8 2->Support for FEAT\_FP8FMA 3->Support for FEAT\_FP8DOT4 4->Support for FEAT\_FP8DOT2.

Type: `uint8_t`

Default value: 0



**fpcr\_short\_vector\_raz**

FPSCR and FPCR fields LEN and STRIDE are hardwired to 0.

Type: `bool`

Default value: `false`

**fpsr\_res0\_stateful\_mask**

Mask for stateful bits of FPSR which are **RES0**.

Type: `uint32_t`

Default value: `0x0`

**fsr\_ext\_bit\_update\_kind**

Set/Clear DFSR/IFSR EA bit on Synchronous/Async External Aborts. 0: Never Set, 1: Set on Synchronous Ext Aborts 2: Set on Asynchronous Ext Aborts 3: Set on both Sync and Async Ext Aborts.

Type: `uint8_t`

Default value: `3`

**gcs\_data\_check\_overrides\_data\_abort**

GCS Data check exceptions are taken before Data Aborts.

Type: `bool`

Default value: `false`

**gcs\_overshoot\_writes**

Number of overshooting GCS records written after a writing a record.

Type: `uint64_t`

Default value: `0x0`

**gcspr\_sync\_immediate**

If true, writing to GCSPR\_ELx registers has immediate effect regardless of `has_delayed_sysreg` flag.

Type: `bool`

Default value: `false`

**gic.GICC-offset**

Offset from PERIPHBASE for GICC registers.

Type: `uint32_t`

Default value: `0x2000`

#### **`gic.GICD-offset`**

Offset from PERIPHBASE for GICD registers. Will be ignored when GICv3 CPU interface is enabled, as distributor is then external to the cluster.

Type: `uint32_t`

Default value: `0x1000`

#### **`gic.GICH-offset`**

Offset from PERIPHBASE for GICH registers.

Type: `uint32_t`

Default value: `0x4000`

#### **`gic.GICH-other-CPU-offset`**

Offset from PERIPHBASE for GICH registers for accessing other CPUs in the cluster. Set to 0 to disable.

Type: `uint32_t`

Default value: `0x5000`

#### **`gic.GICV-alias`**

Offset from PERIPHBASE for alias of GICV registers. When `gicv2-only`, if zero no alias will be created; if `gicv2-only=0`, the param is deprecated, when zero or unset an alias is created in the place mandated by the architecture (`GICV-base+0xF000`).

Type: `uint32_t`

Default value: `0x0`

#### **`gic.GICV-offset`**

Offset from PERIPHBASE for GICV registers.

Type: `uint32_t`

Default value: `0x6000`

#### **`gic.PERIPH-size`**

Size of registers based at PERIPHBASE that are considered to be owned by the GIC. Any accesses in the range PERIPHBASE to PERIPHBASE+`gic.PERIPH-size-1` that do not match GIC registers will be treated as **RAZ/WI**.

Type: `uint32_t`

Default value: `0x8000`

### **`gicv3.A3-affinity-supported`**

Whether a non-zero value for affinity at level 3 is supported.

Type: `bool`

Default value: `false`

### **`gicv3.BPR-min`**

The minimum value for the GICC\_BPR register (non-secure version will be 1 + this value).

Type: `uint8_t`

Default value: `2`

### **`gicv3.EOI-check-CPUID`**

Check CPU ID specified for accesses to EOI registers (rather than just ending highest priority active interrupt).

Type: `bool`

Default value: `false`

### **`gicv3.EOI-check-ID`**

Check Interrupt ID specified for accesses to EOI registers (rather than just ending highest priority active interrupt).

Type: `bool`

Default value: `false`

### **`gicv3.EOI-deactivate-any-interrupt`**

Allow an EOI to deactivate interrupts that aren't the highest priority active interrupt (EOI-ignore-out-of-order must be false otherwise this is ignored).

Type: `bool`

Default value: `false`

### **`gicv3.EOI-ignore-out-of-order`**

Ignore EOI writes that cannot end the highest priority active interrupt.

Type: `bool`

Default value: `true`

**gicv3.FIQEn-RAO**

GICC\_CTLR.FIQEn is read as one, write insensitive.

Type: `bool`

Default value: `false`

**gicv3.IIDR\_base**

The base value for calculating the GICC\_IIDR register value.

Type: `uint32_t`

Default value: `0x43b`

**gicv3.LR-count**

The number of implemented list registers.

Type: `uint8_t`

Default value: `16`

**gicv3.PMHE-RAO-WI**

ICC\_CTLR\_EL\*.PHME is read as one, write insensitive.

Type: `bool`

Default value: `false`

**gicv3.PMHE-RAZ-WI**

ICC\_CTLR\_EL\*.PHME is read as zero, write insensitive.

Type: `bool`

Default value: `false`

**gicv3.PMHE-release-set-packet**

if PHME is enabled, whether a SET packet is released by CPU Intf in Upstream Ack window.

Type: `bool`

Default value: `false`

**gicv3.SRE-EL2-enable-RAO**

When ICC\_SRE\_EL2.SRE is **RAO/WI**, makes ICC\_SRE\_EL2.Enable **RAO/WI**.

Type: `bool`

Default value: `false`

**gicv3.SRE-EL3-enable-RAO**

When ICC\_SRE\_EL3.SRE is **RAO/WI**, makes ICC\_SRE\_EL3.Enable **RAO/WI**.

Type: `bool`

Default value: `false`

**gicv3.SRE-EL3-set-once**

Restrict SRE EL3 to be set only once.

Type: `bool`

Default value: `false`

**gicv3.SRE-enable-action-on-mmap**

Allowed values are: 0-SRE one allows mmap access. 1-SRE one disables mmap access. 2-SRE one makes mmap access **RAZ-WI**.

Type: `uint8_t`

Default value: 0

**gicv3.STATUSR-implemented**

If GICv3 CPU interface is being used, this determines whether the STATUS registers are implemented.

Type: `bool`

Default value: `true`

**gicv3.VBPR-min**

The minimum value for the GICV\_BPR register (non-secure version will be 1 + this value).

Type: `uint8_t`

Default value: 2

**gicv3.VFIQEn-RAO**

ICH\_VMCR\_EL2.VFIQEn is read as one, write insensitive.

Type: `bool`

Default value: `false`

**gicv3.cpuintf-mmap-access-level**

Allowed values are: 0-mmap access is supported for GICC,GICH,GICV registers. 1-mmap access is supported only for GICV registers. 2-mmap access is not supported.

Type: `uint8_t`

Default value: `0`

### **`gicv3.dir-trap-support`**

The cpu supports separate trapping of ICC\_DIR\_EL1 to EL2.

Type: `bool`

Default value: `true`

### **`gicv3.el3_trap_priority_when_secure_debug_disabled`**

Undef to access priorities group register when secure debug is disabled.

Type: `bool`

Default value: `false`

### **`gicv3.extended-interrupt-range-support`**

Device has support for extended SPI/PPI ID ranges.

Type: `bool`

Default value: `false`

### **`gicv3.gicv2-only`**

Limit the GIC implementation to GICv2 features only.

Type: `bool`

Default value: `false`

### **`gicv3.idle-is-ff`**

For GICC/GICV RPR, when idle, return FF when true, minimum supported priority otherwise.

Type: `bool`

Default value: `true`

### **`gicv3.ignore-DIR-write-when-EOImode-not-set`**

Ignore **UNPREDICTABLE** access to GICC\_DIR register.

Type: `bool`

Default value: `true`

### **`gicv3.interrupt-bypass-support`**

Interrupt bypass support, set to false for devices not supporting interrupt bypass.

Type: `bool`

Default value: `true`

### **`gicv3.local-SEIs`**

Generate SEI to signal internal issues.

Type: `bool`

Default value: `false`

### **`gicv3.local-VSEIs`**

Generate VSEI to signal internal issues.

Type: `bool`

Default value: `false`

### **`gicv3.physical-ID-bits`**

Number of physical ID bits implemented.

Type: `uint8_t`

Default value: 16

### **`gicv3.priority-bits`**

Number of priority bits implemented.

Type: `uint8_t`

Default value: 5

### **`gicv3.send-PMHE-command-only-when-priority-changes`**

Send PMHE upstream command to distributor only when write to ICC\_PMR\_EL1 changes the priority.

Type: `bool`

Default value: `false`

### **`gicv3.sgi-range-selector-support`**

Device has support for the Range Selector feature for SGI.

Type: `bool`

Default value: `false`

**gicv3.suppress-virtual-enables-comms**

In GICv3 only mode, prevents the GIC CPUIF from communicating UpstreamWrite/VirtualEnables to the IRI.

Type: `bool`

Default value: `true`

**gicv3.virtual-ID-bits**

Number of virtual ID bits implemented.

Type: `uint8_t`

Default value: 16

**gicv3.virtual-lpi-support**

When GICv3 is supported, indicates a cut down CPUIF interface with no support of VLPI (GICv3 only) when false.

Type: `bool`

Default value: `true`

**gicv3.virtual-priority-bits**

Number of virtual priority bits implemented.

Type: `uint8_t`

Default value: 5

**gicv3.without-DS-support**

GICv3 CPU interfaces do not support disabling security in the distributor (GICD\_CTLR.DS=1).

Type: `bool`

Default value: `false`

**gicv4.mask-virtual-interrupt**

If true, virtual interrupts can be masked from being reported to virtual CPU interface by setting ICH\_HCR\_EL2.DVIM 1. No control otherwise.

Type: `bool`

Default value: `false`

**gicv5.config\_file**

File path for the GICv5 configuration yaml. The file lists the GICv5 params.



Type: `string`

Default value: `N/A`

### **`gicv5.has_gcie_legacy`**

When set to true, FEAT\_GCIE\_LEGACY is supported.

Type: `bool`

Default value: `false`

### **`gicv5.interrupt-bypass-support`**

Interrupt bypass support. when set to true, bypasses GICv5 CPU interface to signal interrupts to the PE.

Type: `bool`

Default value: `false`

### **`global_debug_rom.ROMDEVID`**

Value of Debug Rom Device Identification Register.

Type: `uint64_t`

Default value: `0x0`

### **`global_debug_rom.ROMPIDR`**

Value of Debug Rom Peripheral Identification Register.

Type: `uint64_t`

Default value: `0x4000bb000`

### **`global_debug_rom.ROMPRIDR0`**

Value of Debug ROM Power RequestID Register.

Type: `uint8_t`

Default value: `1`

### **`gpccr_el3_gpcp_behaviour`**

Used to control impdef behaviour when GPCP=1 (0->Faults are always generated and reported, 1->Faults are not generated and reported), 2->Faults are generated and reported only for Arm recommended cases.

Type: `uint8_t`

Default value: `2`

**gpt\_tlb\_size**

Number of separate GPT TLB entries.

Type: `uint32_t`

Default value: `0x0`

**gpt\_walkcache\_size**

Number of GPT walk cache entries.

Type: `uint32_t`

Default value: `0x0`

**hardware\_translation\_table\_update\_implemented**

Implement hardware translation table updates from ARMv8.1 (FEAT\_HAFDBS).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.1 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: `1`

**has-gicv4.1**

GICv4.1 is enabled, and all the features with GICv4.1 are implemented (FEAT\_GICv4p1).

Type: `bool`

Default value: `false`

**has\_128\_bit\_atomic\_instructions**

Implement 128-bit Atomic Instructions (FEAT\_LSE128); mandatory in the presence of Future Architecture Technologies (FAT)values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: `0`

**has\_128\_bit\_tt\_descriptors**

Implement 128-bit Translation Table Descriptors (FEAT\_D128)values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: `0`

**has\_16bit\_asids**

Enable 16-bit ASIDs; mandatory in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**has\_16bit\_vmids**

Implement support for 16-bit VMIDs from ARMv8.1 (FEAT\_VMID16).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.1 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: `1`

**has\_16k\_granule**

Implement the 16k LPAE translation granule.

Type: `bool`

Default value: `false`

**has\_4k\_granule**

Implement the 4k LPAE translation granule.

Type: `bool`

Default value: `true`

**has\_52bit\_address\_with\_16k**

Implements Armv8.7 52-bit IPA/PA support for 16k (FEAT\_LPA2).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.7 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: `0`

**has\_52bit\_address\_with\_4k**

Implements Armv8.7 52-bit IPA/PA support for 4k (FEAT\_LPA2).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.7 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: `0`

**has\_56\_bit\_va**

56-bit Physical Address, identified as (FEAT\_LVA3)values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

**has\_64bit\_pmu\_ext\_access**

Implement 64-bit pmu external interface access (FEAT\_PMU\_EXT64)values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.8 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

**has\_64k\_granule**

Implement the 64k LPAE translation granule.

Type: `bool`

Default value: `true`

**has\_aarch32\_dbgdidr\_etc**

DBGDIDR, DBGDRAR, DBGDSAR exist even if EL1 doesn't implement AArch32.

Type: `bool`

Default value: `true`

**has\_aarch32\_hpd**

If true then hierarchical permission disable is supported in AArch32 (FEAT\_AA32HPD).

Type: `bool`

Default value: `false`

**has\_aarch64**

All implemented exception levels can run in AArch64.

Type: `bool`

Default value: `true`

**has\_actlr2**

If true ACLTR2 exists and ACTLR2(NS) is aliased to ACTLR\_EL1[63:32].

Type: `bool`

Default value: `false`

### **has\_actlr\_virtualisation**

If true ACTLR\_EL12 is implemented and ACTLR\_EL1 supports virtualisation.

Type: `bool`

Default value: `false`

### **has\_address\_breakpoint\_linking**

Implement Address Breakpoint Linking Extension (FEAT\_ABLE) values of this parameter are:-  
0, feature is not enabled.- 1, feature is implemented if ARMv9.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **has\_afgdt\_virtualisation**

If true AFGDTpn\_EL12 is implemented and AFGDTpn\_EL1 supports virtualisation.

Type: `bool`

Default value: `true`

### **has\_amu**

Implement activity monitor functionality from ARMv8.4 (FEAT\_AMUv1). values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **has\_amu\_ext64**

Implement 64-bit external interface to the Activity Monitors (FEAT\_AMU\_EXT64). values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **has\_amu\_extacr**

If true then implement AMU external access control registers (FEAT\_AMU\_EXTACR).

Type: `bool`

Default value: `false`

#### **has\_arm\_v8-1**

Implement the ARMv8.1 Extension.

Type: `bool`

Default value: `false`

#### **has\_arm\_v8-2**

Implement the ARMv8.2 Extension.

Type: `bool`

Default value: `false`

#### **has\_arm\_v8-3**

Implement the ARMv8.3 Extension.

Type: `bool`

Default value: `false`

#### **has\_arm\_v8-4**

Implement the ARMv8.4 Extension.

Type: `bool`

Default value: `false`

#### **has\_arm\_v8-5**

Implement the ARMv8.5 Extension.

Type: `bool`

Default value: `false`

#### **has\_arm\_v8-6**

Implement the ARMv8.6 Extension.

Type: `bool`

Default value: `false`

#### **has\_arm\_v8-7**

Implement the Armv8.7 Extension.

Type: `bool`

Default value: `false`

### **has\_arm\_v8-8**

Implement the ARMv8.8 Extension.

Type: `bool`

Default value: `false`

### **has\_arm\_v8-9**

Implement the ARMv8.9 Extension.

Type: `bool`

Default value: `false`

### **has\_arm\_v9-0**

Implement the ARMv9.0 Extension.

Type: `bool`

Default value: `false`

### **has\_arm\_v9-1**

Implement the ARMv9.1 Extension.

Type: `bool`

Default value: `false`

### **has\_arm\_v9-2**

Implement the ARMv9.2 Extension.

Type: `bool`

Default value: `false`

### **has\_arm\_v9-3**

Implement the ARMv9.3 Extension.

Type: `bool`

Default value: `false`

### **has\_arm\_v9-4**

Implement the ARMv9.4 Extension.

Type: `bool`

Default value: `false`

### **has\_arm\_v9-5**

Implement the ARMv9.5 Extension.

Type: `bool`

Default value: `false`

### **has\_arm\_v9-6**

Implement the ARMv9.6 Extension.

Type: `bool`

Default value: `false`

### **has\_arm\_v9-7**

Implement the ARMv9.7 Extension.

Type: `bool`

Default value: `false`

### **has\_asid2**

If true then support for use of two concurrent ASIDs (FEAT\_ASID2) values of this parameter are:- 1, feature is implemented if ARMv9.5 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_at\_with\_pan**

Implement new AT instructions with PAN support (FEAT\_PAN2). values of this parameter are:- 1, feature is implemented if ARMv8.2 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_ats1a**

Support for ATS1ExR instructions (FEAT\_ATS1A) values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0



**has\_attribute\_index\_enhancement**

Memory Attribute Index Enhancement (FEAT\_AIE) values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

**has\_axflag\_xaflag**

Implement flag manipulation instructions (AXFlag, XAFlag) from ARMv8.5 (FEAT\_FlagM2). values of this parameter are:- 1, feature is implemented if ARMv8.5 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

**has\_axflag\_xaflag\_frint**

Implement flag manipulation instructions (AXFlag, XAFlag) and floating-point rounding to int instructions (FRINT[32|64][X|Z]) from ARMv8.5. If this parameter is enabled, it also enables both `has_axflag_xaflag` and `has_frint`. If support for only one of the features is needed, please use the individual parameters and do not enable this one (FEAT\_FlagM2, FEAT\_FRINTTS). values of this parameter are:- 1, feature is implemented if ARMv8.5 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

**has\_bc**

Implement Armv8.8 Hinted Conditional Branch (FEAT\_HBC) values of this parameter are:- 1, feature is implemented if ARMv8.8 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

**has\_branch\_target\_exception**

Implement Branch target identification mechanism from ARMv8.5 (FEAT\_BTI) (DEPRECATED: Use `bti_support_level` instead). values of this parameter are:- 1, feature is implemented if ARMv8.5 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

**has\_brbe**

If true, implements branch record buffer extension (FEAT\_BRBE).

Type: `bool`

Default value: `false`

### **has\_brbe\_v1p1**

If true, implements FEAT\_BRBEv1p1.

Type: `bool`

Default value: `false`

### **has\_ccidx**

Implement the ARMv8.3 FEAT\_CCIDR Extension. Extending the ccidr number of sets (FEAT\_CCIDX).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.3 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **has\_cfinv\_rmif\_setf**

Implement flag manipulation (CFINV, RMIF, SETF8, SETF16) instructions from ARMv8.4 (FEAT\_FlagM).values of this parameter are:- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_cflt**

Support for compare and fault instructions (FEAT\_CFLT).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.7 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **has\_chkfeat**

Implement CHKFEAT instruction from ARMv9.4 (FEAT\_CHK).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **has\_clear\_bhb**

Implement Clear Branch History information instruction (FEAT\_CLRBHB).values of this parameter are:- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_clear\_other\_speculation\_by\_context**

Implement execution and data prediction invalidation from Armv8.9 (FEAT\_SPECRES2).values of this parameter are:- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_cmh**

Implement Coherency scalability hints (FEAT\_CMH)values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.7 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **has\_cmo\_wr\_control**

Whether stage1/2 CMO write perm control is supported (FEAT\_CMOW)values of this parameter are:- 1, feature is implemented if ARMv8.8 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_cmpbr**

Implement compare and branch instructions from ARMv9.6, optional in ARMv9.5 (FEAT\_CMPBR)values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.5 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **has\_coherent\_icache**

Whether icache invalidation to the point of unification is required for instruction to data coherence.  
true - Invalidate operations not required.

Type: `bool`

Default value: `false`

**has\_common\_not\_private\_translations**

Implement the TTBRn\_ELx.CnP (Common not Private) controls from ARMv8.2 (FEAT\_TTCNP).values of this parameter are:- 1, feature is implemented if ARMv8.2 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

**has\_complex\_number**

Implement ARMv8.3 complex number support, Multiply Accumulate and Add instructions (FEAT\_FCMA).values of this parameter are:- 1, feature is implemented if ARMv8.3 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

**has\_const\_pac**

Feature for singular selection of PAC field (FEAT\_CONSTPACFIELD).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.8 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

**has\_cssc**

Support for common short sequence compression instructions (FEAT\_CSSC).values of this parameter are:- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

**has\_cvadp\_support**

Implement instruction to support cache clean by deep persistence (DC CVADP) from ARMv8.5, can be selected for core implemented on any arch version starting ARMv8.2 (FEAT\_DPB, FEAT\_DPB2).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.2 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

**has\_data\_abort\_syndrome\_enhancements**

Support for Data Abort syndrome information enhancements (FEAT\_EAESR). Values of this parameter are: - 1, feature is implemented if ARMv9.7 is enabled. - 2, feature is implemented.

Type: `uint8_t`

Default value: 1

**has\_data\_alignment\_flag**

Implement non-optimal misalignment flag for PMU/SPE from ARMv8.5. Values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.5 is enabled. - 2, feature is implemented.

Type: `uint8_t`

Default value: 0

**has\_debug\_rom**

If true, a debug ROM will be generated describing the cluster's debug components.

Type: `bool`

Default value: `true`

**has\_delayed\_brbe\_records**

If true, a synchronization barrier is required to update the BRB records (FEAT\_BRBE).

Type: `bool`

Default value: `true`

**has\_delayed\_ctireg**

Delay the functional effect of CTI register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**has\_delayed\_mdscr\_el1**

Delay the functional effect of MDSCR\_EL1 register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

### **has\_delayed\_oslar\_el1**

Delay the functional effect of OSLAR\_EL1 register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

### **has\_delayed\_pmureg**

Delay the functional effect of PMU register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

### **has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

### **has\_delayed\_wfe\_trap**

Implements Configurable Delayed WFE trapping from ARMv8.6 (FEAT\_TWED).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.6 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **has\_dgh**

Implements Data Gathering Hint instruction from ARMv8.6 (FEAT\_DGH).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.6 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_dot\_product**

Implement the dot product (UDOT, SDOT) instructions from ARMv8.4 (FEAT\_DotProd).values of this parameter are:- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_e0pd**

Implement ARMv8.5 feature to prevent unprivileged access to one half of the memory (FEAT\_EOPD).values of this parameter are:- 1, feature is implemented if ARMv8.5 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

### **has\_e2h\_rao**

Whether the implementation treats HCR\_EL2.E2H as Read-As-One (**RAO**). 0 : FEAT\_E2H0 implemented.values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.1 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

### **has\_ebep**

Implement Exception-Based Event Profiling from ARMv9.4 (FEAT\_EBEP).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.4 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

### **has\_ebf16**

Support for Extended BFloat16 Behaviours (FEAT\_EBF16).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.2 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

### **has\_ecbhb**

Implement Exploitative Control using Branch History information between exception levels (FEAT\_ECBHB).values of this parameter are:- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

### **has\_edacr**

Implement EDACR register.

Type: `bool`

Default value: `true`

### **has\_edhshr**

Implement external debug halt status register (FEAT\_EDHSHR). 0: FEAT\_EDHSHR is not implemented unless architecturally required by another feature, 1: FEAT\_EDHSHR is implemented, 2: FEAT\_EDHSHR is implemented (extends EDHSHR to include the VNCR, CM, and WnR fields), 0xF: FEAT\_EDHSHR implementation is dependent on FEAT\_SME.

Type: `uint8_t`

Default value: 15

### **has\_el2**

Implements EL2.

Type: `bool`

Default value: `true`

### **has\_el3**

Implements EL3.

Type: `bool`

Default value: `true`

### **has\_enhanced\_pac**

If pointer authentication is enabled then implement enhanced PAC (FEAT\_EPAC).

Type: `bool`

Default value: `false`

### **has\_enhanced\_pan**

Implements Armv8.7 Enhanced PAN feature (FEAT\_PAN3) values of this parameter are:- 1, feature is implemented if ARMv8.7 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_enhanced\_software\_step**

Implement Enhanced Software Step Extension (FEAT\_STEP2) values of this parameter are:- 1, feature is implemented if ARMv9.5 is enabled.- 2, feature is implemented.

Type: `uint8_t`



Default value: 1

**has\_ete**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (-plugin or -P).

Type: bool

Default value: false

**has\_ets**

Whether Enhanced Translation Synchronization is supported. (NOTE: DEPRECATED: use ets\_level instead).

Type: bool

Default value: false

**has\_exception\_trapping\_form\_of\_vector\_catch**

Implement the exception trapping form of vector catch debug event.

Type: bool

Default value: true

**has\_extended\_recip\_estimate**

Implements increased precision of reciprocal instructions (FEAT\_RPRES).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.7 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

**has\_external\_rndr**

Implement external random number generator module. When enabling this with has\_rndr enabled, the external random number generator will be used instead of internal random number generator.values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.5 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

**has\_f16f32dot**

Implement FEAT\_F16F32DOT.

Type: bool

Default value: `false`

**has\_f16f32mm**

Implement FEAT\_F16F32MM.

Type: `bool`

Default value: `false`

**has\_f16mm**

Implement FEAT\_F16MM.

Type: `bool`

Default value: `false`

**has\_f8f16mm**

Implement FEAT\_F8F16MM and dependent features.

Type: `bool`

Default value: `false`

**has\_f8f32mm**

Implement FEAT\_F8F32MM and dependent features.

Type: `bool`

Default value: `false`

**has\_faminmax**

Implement FEAT\_FAMINMAX.

Type: `bool`

Default value: `false`

**has\_far\_not\_valid**

Implements FnV bit in ESR\_ELx and xFSR, FAR not valid for synchronous external aborts.

Type: `bool`

Default value: `false`

**has\_far\_not\_valid\_dfsc**

Implements FnV bit in ESR\_ELx, FAR not valid for synchronous external aborts for Data Abort.

Type: `bool`

Default value: `false`

### **has\_far\_not\_valid\_ifsc**

Implements FnV bit in ESR\_ELx and xFSR, FAR not valid for synchronous external aborts for Instruction Abort.

Type: `bool`

Default value: `false`

### **has\_fdit**

Support higher exception levels to enforce data-independent timing (FEAT\_FDIT).values of this parameter are:- 1, feature is implemented if ARMv9.7 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_feat\_pops**

Whether ARMv9.6 RAS support for clean-and-invalidate of data by virtual address to Point of Physical Storage (FEAT\_PoPS)values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.6 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **has\_fgt**

Implements Fine-grained Virtualization Traps extension from ARMv8.6 (FEAT\_FGT).values of this parameter are:- 1, feature is implemented if ARMv8.6 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_fgt2**

Implement additional FGT traps introduced in ARMv8.9 (FEAT\_FGT2).values of this parameter are:- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_fgwt3**

If true then Fine Grained Write EL3 is enabled (FEAT\_FGWTE3).

Type: `bool`

Default value: `false`

### **has\_fixed\_function\_instr\_counter**

Implement fixed-function instruction counter (FEAT\_PMUv3\_ICNTR) values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **has\_fp16**

Implement the half-precision floating-point data processing instructions from ARMv8.2 (FEAT\_FP16). values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.2 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_fp16\_fmlal**

Implement the New Floating Point Multiplication Variant (FP16 FMLAL, FMLSL) instructions from ARMv8.4. Only supported if `has_fp16=0x1` (FEAT\_FHM). values of this parameter are:- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_fpmr**

Implement FPMR (FEAT\_FPMR).

Type: `bool`

Default value: `false`

### **has\_fprcvt**

Implement FEAT\_FPRCVT FP convert instructions from ARMv9.6, optional in ARMv9.5 (FEAT\_FPRCVT) values of this parameter are:- 1, feature is implemented if ARMv9.6 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

**has\_frint**

Implement floating-point rounding to int instructions (FRINT[32|64][X|Z]) from ARMv8.5 (FEAT\_FRINTTS).values of this parameter are:- 1, feature is implemented if ARMv8.5 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

**has\_gcs**

Implement Guarded Control Stack Extension from ARMv9.4 (FEAT\_GCS).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.4 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

**has\_generic\_authentication**

Implement ARMv8.3 generic authentication.values of this parameter are:- 1, feature is implemented if ARMv8.3 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

**has\_guest\_translation\_granule**

Implement mechanism for guest translation granule identification from ARMv8.5, ID values determined by stage1 granule configuration parameters (FEAT\_GTG).values of this parameter are:- 1, feature is implemented if ARMv8.5 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

**has\_haft**

Implement Hardware managed Access Flag for Table Descriptors (FEAT\_HAFT).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.7 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

**has\_hardware\_accelerator\_for\_cleaning\_dirty\_state**

Whether hardware accelerator for cleaning Dirty state is supported (FEAT\_HACDBS).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.5 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

**has\_hardware\_dirty\_state\_tracking\_structure**

Whether hardware Dirty state tracking Structure is supported (FEAT\_HDBSS).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.5 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

**has\_hardware\_translation\_table\_update**

Type of hardware translation table supported (when enabled by hardware\_translation\_table\_update\_implemented). 0, not implemented. 1, access bit updates implemented. 2, access bit updates and dirty bit mechanism implemented (FEAT\_HAFDBS).

Type: uint8\_t

Default value: 2

**has\_hcrx\_el2**

Implements new HCRX\_EL2 id register from Armv8.7 (FEAT\_HCX).values of this parameter are:- 1, feature is implemented if ARMv8.7 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

**has\_hpmn0**

Allow hypervisor to set MDCR\_EL2.HPMN to 0 (FEAT\_HPMN0).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.8 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

**has\_id\_reg\_read**

Implement read access to the ID registers (ESR\_ELx.EC=0x18) (FEAT\_IDST).values of this parameter are:- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_iesb**

Implement support for implicit error sync event from ARMv8.2 (FEAT\_IESB).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.2 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_itd**

Implement the optional IT disable feature.

Type: `bool`

Default value: `true`

### **has\_ite**

Implement Instrumentation Trace Extension from ARMv9.4 (FEAT\_ITE).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **has\_jscvt**

Implement ARMv8.3 javascript Floating-point to Integer conversion instruction (FEAT\_JSCVT).values of this parameter are:- 1, feature is implemented if ARMv8.3 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_large\_system\_ext**

Implement the ARMv8 Large System Extensions (FEAT\_LSE).

Type: `bool`

Default value: `false`

### **has\_large\_ttbr\_ba\_without\_lpa**

When FEAT\_LPA is not implemented, whether TTBR base address supports large values (52 bits) or not (48 bits).

Type: `bool`

Default value: `true`

### **has\_large\_va**

Implement support for the extended 52-bit virtual addresses from ARMv8.2 (FEAT\_LVA).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.2 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **has\_ldapur\_stlur**

Implement support for LDAPR and STLUR instructions with immediate offsets from ARMv8.4 (FEAT\_LRCPC2).values of this parameter are:- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_ldm\_stm\_ordering\_control**

Implement the SCTLRL\_ELx.LSMAOE (Load/Store Multiple Atomicity and Ordering Enable) and SCTLRL\_ELx,nTLSMD (no Trap Load/Store Multiple to Device) controls from ARMv8.2 (FEAT\_LSMAOC).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.2 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **has\_lorrl**

Support for Limited Order Regions in Realm PA space (FEAT\_LORRL).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.7 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **has\_lrcpc**

If true then it support the RCpc feature from ARMv8.3 (FEAT\_LRCPC).

Type: `bool`

Default value: `false`



**has\_lrcpc3**

Implement Release Consistency processor consistent (RCpc) feature from Armv8.9

(FEAT\_LRCPC3).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

**has\_lscp**

Implement 128-bit Load acquire and store release pair single-copy atomic instructions

(FEAT\_LSCP).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.7 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

**has\_lsfe**

Implement A64 base Atomic floating-point in-memory instructions (FEAT\_LSFE).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.3 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

**has\_lsui**

Implement additional load and store unprivileged instructions (FEAT\_LSUI).values of this parameter are:- 1, feature is implemented if ARMv9.6 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

**has\_lut**

Implement FEAT\_LUT.

Type: `bool`

Default value: `false`

**has\_mbist\_never1\_ae25**

Implements the MBIST Never1 detection for AE25 class cpu.

Type: `bool`

Default value: `false`

**has\_mismatch\_and\_range\_breakpoints**

Implement Mismatch and Range Breakpoints (FEAT\_BWE) values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.4 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

**has\_mismatch\_watchpoints**

Implement Breakpoints and Watchpoints Enhancements (FEAT\_BWE2) values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.5 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

**has\_mops\_go\_option**

Implement MTE Tag set only MOPS instructions (FEAT\_MOPS\_GO). 0, not implemented. 1, implemented using Option A. 2, implemented using Option B.

Type: uint8\_t

Default value: 0

**has\_mops\_option**

Implement Armv8.8 standard instructions for memory operations (FEAT\_MOPS). 0, not implemented (unsupported if Armv8.8 is enabled). 1, implemented using Option A. 2, implemented using Option B.

Type: uint8\_t

Default value: 0

**has\_mpam**

Implement ARMv8.4 MPAM Registers and associated functionality (FEAT\_MPAM) values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

**has\_mpamv2**

Implement MPAM architecture 2.0 (FEAT\_MPAMv2) values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.7 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

### **has\_mpamv2\_alt\_id**

Implement alternative identifier for MPAM architecture 2.0 (FEAT\_MPAMv2\_ALT\_ID).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.7 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

### **has\_mpamv2\_instr\_alt\_id**

Implement alternative identifier for instruction fetches when FEAT\_MPAMv2\_ALT\_ID is enabled.values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.7 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

### **has\_mpamv2\_vid**

Implement identifier virtualization for MPAM architecture 2.0 (FEAT\_MPAMv2\_VID).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.7 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

### **has\_mpm**

Implement max-power mitigation mechanism (MPMM).

Type: bool

Default value: false

### **has\_mt\_pmu\_disable\_feature**

Implements Multi-threading PMU disable extension from ARMv8.6 (FEAT\_MTPMU). 0: FEAT\_MTPMU is disabled, 1: FEAT\_MTPMU is enabled if ARMv8.6 is implemented, 2: FEAT\_MTPMU is cherry-picked, 0xF: The feature is disabled and is represented by value 0xF in ID\_AA64DFR0\_EL1.MTPMU.

Type: uint8\_t

Default value: 0

### **has\_mte\_async\_faults**

Whether MTE asynchronous faults are supported (FEAT\_MTE\_ASYNC).

Type: `bool`

Default value: `true`

### **has\_mte\_eirg**

Implement Enhanced Insert Random Tag (FEAT\_MTE\_EIRG) values of this parameter are:- 1, feature is implemented if ARMv9.7 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_mte\_fgt**

When FEAT\_VMTE or FEAT\_MTE2 are implemented, whether FEAT\_MTEFGT is implemented.

Type: `bool`

Default value: `false`

### **has\_mte\_perm**

Implement tag access permission (FEAT\_MTE\_PERM).

Type: `bool`

Default value: `false`

### **has\_mte\_tag\_related\_fault\_high\_prio\_than\_data**

For DC GZVA, Whether MMU faults generated by tag access has higher priority than faults due to data access.

Type: `bool`

Default value: `false`

### **has\_nested\_virtualization**

Implement ARMv8.3 nested virtualization (FEAT\_NV) values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.3 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_nmi**

Implement AARCH64 Non-Maskable Interrupts (FEAT\_NMI) values of this parameter are:- 1, feature is implemented if ARMv8.8 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_no\_os\_double\_lock**

Do not implement the OS double-lock (FEAT\_DoubleLock).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

### **has\_non\_context\_synchronizing\_exception\_controls**

Implement cosmetic controls for whether exception entry and exit are context synchronizing events (SCTLR\_ELx.{EIS,EOS}) from ARMv8.5 (FEAT\_ExS).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.5 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

### **has\_nv1\_raz**

Whether the implementation treats HCR\_EL2.NV1 as Read-As-Zero (**RAZ**), if has\_e2h\_rao = 1.values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.3 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

### **has\_nv\_frac**

Whether the NV\_frac behavior is supported. (DEPRECATED: use nv\_frac\_support\_level)values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

### **has\_occmo**

Implement The DC CIVAOC instruction (FEAT\_OCCMO)values of this parameter are:- 1, feature is implemented if ARMv9.6 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

### **has\_par\_bit10\_razwi**

Whether PAR\_EL1[10] is **RAZ/WI**.

Type: `bool`

Default value: `false`

### **has\_partial\_delayed\_mdscr\_el1**

has\_delayed\_oslar\_el1 only apply to some bits of MDSCR\_EL1 (MDE, KDE, TDCC, SS).

Type: `bool`

Default value: `false`

### **has\_pauth\_enhctl**

Support for Enhanced PAC controls (FEAT\_PAuth\_EnhCtl) values of this parameter are:- 1, feature is implemented if ARMv9.7 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_pc\_sample\_based\_profiling**

If true, pc sample-based profiling is enabled (FEAT\_PCSRv8, FEAT\_PCSRv8p2).

Type: `bool`

Default value: `true`

### **has\_pc\_sample\_profiling\_enable**

Whether PC Sample profiling enable is implemented (FEAT\_PCSRv8p9).

Type: `bool`

Default value: `false`

### **has\_pcdphint**

Support for producer-consumer data placement hints instructions (FEAT\_PCDPHINT) values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.6 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **has\_per\_cluster\_debug\_auth\_ports**

If true then the debug authentication ports i.e. spniden, niden, rpliden, rtpiden, dbgen, spiden are available per cluster.

Type: `bool`

Default value: `false`

### **`has_permission_indirection_s1`**

Implement the Permission Indirection Extension at stage 1 (FEAT\_S1PIE) values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **`has_permission_indirection_s2`**

Implement the Permission Indirection Extension at stage 2 (FEAT\_S2PIE) values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **`has_permission_overlay_s1`**

Implement the Permission Overlay Extension at stage 1 (FEAT\_S1POE) (NOTE: DEPRECATED: Use `s1poe_support_level` instead) values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **`has_permission_overlay_s2`**

Implement the Permission Overlay Extension at stage 2 (FEAT\_S2POE) values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **`has_plb_conflict_abort`**

Constrained unpredictable when PLB conflicts observed, True : PLB conflict abort reported.

Type: `bool`

Default value: `false`

**has\_pmss**

Implement PMU Snapshot Extension from Armv8.9 (FEAT\_PMUv3\_SS).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

**has\_pmu**

Implement the optional Performance Monitors Extension (FEAT\_PMUv3). 0, Not Implemented. 1, Implemented. 2, PMU is IMPLEMENTATION\_DEFINED, PMU version would be set to 0xF and would behave as if no PMU is implemented.

Type: uint8\_t

Default value: 1

**has\_pmu\_edge\_detection**

Implement PMU Event edge detection (FEAT\_PMUv3\_EDGE)values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

**has\_pmu\_extpmn**

Implement optional PMU extension feature to reserve event counters for external agents from ARMv9.5 (FEAT\_PMUv3\_EXTPMN). 0 not implemented, 1 implementedvalues of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.5 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

**has\_pmu\_for\_sme\_extension**

Implement PMUv3 for Scalable Matrix Extension (SME) from ARMv9.5 (FEAT\_PMUv3\_SME)values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.5 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0



**has\_pmu\_threshold\_linking\_control**

Implement PMU threshold linking control (FEAT\_PMUV3\_TH2) values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.5 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

**has\_pointer\_authentication**

Implement ARMv8.3 pointer authentication (FEAT\_PAuth) values of this parameter are:- 1, feature is implemented if ARMv8.3 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

**has\_prediction\_invalidation\_instructions**

Implement execution and data prediction invalidation from ARMv8.5 (FEAT\_SPECRES) values of this parameter are:- 1, feature is implemented if ARMv8.5 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

**has\_prfm\_slc**

Implement PRFM with SLC hint (FEAT\_PRFM\_SLC) values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

**has\_pstate\_dit**

Implement timing insensitivity of data processing instructions from ARMv8.4 (FEAT\_DIT) values of this parameter are:- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

**has\_pstate\_pan**

Implement the PSTATE.PAN (Privileged Access Never) control from ARMv8.1 (FEAT\_PAN) values of this parameter are:- 1, feature is implemented if ARMv8.1 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

**has\_pstate\_uao**

Implement the PSTATE.UAO (User Access Override) control from ARMv8.2 (FEAT\_UAO).values of this parameter are:- 1, feature is implemented if ARMv8.2 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

**has\_qarma3\_pac**

Supports QARMA3 pointer authentication algorithm (FEAT\_PACQARMA3).

Type: `bool`

Default value: `false`

**has\_ras**

Implements the ARMv8 RAS Extension. 0 = NO\_RAS, 1 = MINIMAL\_RAS, 2 = FULL\_RAS (FEAT\_RAS).

Type: `uint8_t`

Default value: 0

**has\_ras\_aderr**

Implement RAS Asynchronous Device Read Error from Armv8.9 (FEAT\_ADERR).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

**has\_ras\_anerr**

Implement RAS Asynchronous Normal Read Error from Armv8.9 (FEAT\_ANERR).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

**has\_ras\_armv84\_extension**

Implement ARMv8.4 RAS Extension (FEAT\_RASv1p1).values of this parameter are:- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

**has\_ras\_armv89\_double\_fault**

Implement RAS Double Fault Extension from Armv8.9 (FEAT\_DoubleFault2).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

**has\_ras\_armv89\_extension**

Implement RAS extension from Armv8.9 (FEAT\_RASv2).values of this parameter are:- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

**has\_ras\_critical\_error**

[DEPRECATED: Set CI field on first register in error\_record\_feature\_register JSON instead]

ARMv8.4 AArch64 RAS Critical Error is implemented or not. 0 - Feature Not Supported, 1 - Feature always enabled, 2 - Feature is controllable.

Type: uint8\_t

Default value: 0

**has\_ras\_delegated\_serror\_exceptions\_for\_el3**

Implement Delegated SError exceptions for EL3 (FEAT\_E3DSE).values of this parameter are:- 1, feature is implemented if ARMv9.5 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

**has\_ras\_double\_fault**

Implement ARMv8.4 RAS Double Fault Extension (FEAT\_DoubleFault).values of this parameter are:- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

**has\_ras\_fault\_injection**

[DEPRECATED: Set INJ field on first register in error\_record\_feature\_register JSON instead]

Implement ARMv8.4 Standard Fault Injection mechanism.values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

### **has\_ras\_mmmap\_view**

Implement memory mapped view of RAS Registers for cores.

Type: `bool`

Default value: `false`

### **has\_ras\_pfar**

Implement RAS Physical Fault Address Registers from Armv8.9 (FEAT\_PFAR).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **has\_ras\_timestamp**

[DEPRECATED: Set TS field on first register in error\_record\_feature\_register JSON instead]  
ARMv8.4 AArch64 RAS Timestamp register is implemented or not. 0 - No Timestamp is recorded, 1 - Generic Timer timestamp is recorded, 2 - IMP DEF timestamp is recorded.

Type: `uint8_t`

Default value: 0

### **has\_rassa\_acr**

Implement RAS System Architecture v2 optional access control register (FEAT\_RASSA\_ACR).

Type: `bool`

Default value: `false`

### **has\_restriction\_on\_speculative\_data\_loaded**

Implements the ARMv8.5 security feature (Restrictions on the effects of speculation) (FEAT\_CSV3).values of this parameter are:- 1, feature is implemented if ARMv8.5 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_rme**

If true, implements full realm management extension (FEAT\_RME). Note: This parameter is deprecated and will be removed in future releases, please use `rme_support_level` parameter.

Type: `bool`

Default value: `false`

### **has\_rme\_gdi**

Support for RME granular data isolation (FEAT\_RME\_GDI) values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.6 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **has\_rme\_gpc2**

If true then RME GPC2 extension is enabled (FEAT\_RME\_GPC2).

Type: `bool`

Default value: `false`

### **has\_rme\_gpc3**

If true then RME GPC3 extension is enabled (FEAT\_RME\_GPC3).

Type: `bool`

Default value: `false`

### **has\_rndr**

Implement random number instructions to read from RNDR and RNDRSS random number registers from ARMv8.5 (FEAT\_RNG). values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.5 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **has\_rndr\_trap**

Implement trapping for RNDR and RNDRSS random number registers from ARMv8.8. (FEAT\_RNG\_TRAP) values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.8 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **has\_rounding\_doubling\_multiply\_add\_subtract**

Implement the rounding doubling multiply add and subtract instructions from ARMv8.1 (FEAT\_RDM). values of this parameter are:- 1, feature is implemented if ARMv8.1 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

**has\_rprfm**

Support for RPRFM hint instruction (FEAT\_RPRFM).

Type: bool

Default value: false

**has\_sbistc\_ae25**

Implements the SBISTC for AE25 class cpu.

Type: bool

Default value: false

**has\_scr2**

Support for SCR2\_EL3 register (FEAT\_SCR2) values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.7 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

**has\_sctlr2**

Implement SCTL2\_ELx registers (FEAT\_SCTL2) values of this parameter are:- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

**has\_sebep**

Implement Synchronous-Exception-Based Event Profiling from ARMv9.4 (FEAT\_SEBEP). values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.4 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

**has\_secure\_el2**

Implement support for Secure EL2 (FEAT\_SEL2). values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

**has\_self\_hosted\_trace\_extension**

Implement support for the Self-hosted Trace Extensions from ARMv8.4 (FEAT\_TRF).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

**has\_small\_page\_table**

Implement small page table support which increases the maximum value of TxSZ field from ARMv8.4 (FEAT\_TTST). Note: will be unimplemented only if both has\_small\_page\_table=0x0 and has\_secure\_el2=0x0.values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

**has\_software\_lock**

Implement software lock in memory-mapped CTI, PMU, and external debug interfaces.

Type: bool

Default value: true

**has\_spe\_eft**

Implement SPE extended operation type filtering from ARMv9.5 (FEAT\_SPE\_EFT)values of this parameter are:- 1, feature is implemented if ARMv9.5 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

**has\_spe\_fds**

Implement SPE filter by data source from ARMv8.9 (FEAT\_SPE\_FDS)values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

**has\_spe\_for\_sme\_extension**

Implement support of SME to SPE from ARMv9.5 (FEAT\_SPE\_SME).values of this parameter are:- 1, feature is implemented if ARMv9.5 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

### **has\_spe\_fpf**

Implement SPE operation type extension for ASIMD and FP from ARMv9.5 (FEAT\_SPE\_FPF) values of this parameter are:- 1, feature is implemented if ARMv9.5 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_spe\_nvm**

Implement Statistical Profiling physical address mode (FEAT\_SPE\_nVM) values of this parameter are:- 1, feature is implemented if ARMv9.6 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_speculation\_barrier\_inst**

Implement speculation barrier instruction (SB) from ARMv8.5 (FEAT\_SB) values of this parameter are:- 1, feature is implemented if ARMv8.5 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_speculative\_sei**

If true, the PE can generate SError interrupt exceptions from speculative reads of memory, including speculative instruction fetches.

Type: `bool`

Default value: `false`

### **has\_srmask**

Implement bitwise write masks for EL1 control registers (FEAT\_SRMASK) values of this parameter are:- 1, feature is implemented if ARMv9.6 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_srmask2**

Implement bitwise write masks for HCR(X)EL2 registers (FEAT\_SRMASK2) values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.7 is enabled.- 2, feature is implemented.

Type: `uint8_t`



Default value: 0

### **has\_stage2\_ap\_speculative\_update**

Speculative update of S2 AP bit on S1 TTW. 0 = No Update, 1 = Update, 2 = Update including for AT ops.

Type: `uint8_t`

Default value: 0

### **has\_stage2\_fwb**

Implement HCR\_EL2.FWB, stage 2 control of memory types and cacheability (FEAT\_S2FWB) values of this parameter are:- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_stage2\_xnx**

Implement the extended XN[1:0] stage 2 control from ARMv8.2 (FEAT\_XNX). values of this parameter are:- 1, feature is implemented if ARMv8.2 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_stage2\_xnx\_in\_aarch32**

Implement the extended XN[1:0] stage 2 control from ARMv8.2 for Aarch32 (FEAT\_XNX).

Type: `bool`

Default value: `true`

### **has\_statistical\_profiling**

Whether Statistical Based Profiling is implemented (FEAT\_SPE).

Type: `bool`

Default value: `true`

### **has\_supersections**

Whether VMSAv8-32 supersection to support more than 32-bit PA using short descriptor is implemented.

Type: `bool`

Default value: `true`

**has\_sve**

Whether SVE is implemented (FEAT\_SVE). Note: this is required to enable SME (FEAT\_SME) with `sve.has_sme=1`. An SME only implementation can be enabled by setting both as well as `sve.sme_only=1`.

Type: `uint8_t`

Default value: `0`

**has\_synchronous\_load\_atomics**

Report asynchronous abort due to unsupported load atomics as synchronous (Cacheable).

Type: `bool`

Default value: `true`

**has\_synchronous\_load\_atomics\_noncacheable**

Report asynchronous abort due to unsupported load atomics as synchronous (Non-Cacheable).

Type: `bool`

Default value: `true`

**has\_synchronous\_store\_atomics**

Report asynchronous abort due to unsupported store atomics as synchronous (Cacheable).

Type: `bool`

Default value: `false`

**has\_synchronous\_store\_atomics\_noncacheable**

Report asynchronous abort due to unsupported store atomics as synchronous (Non-Cacheable).

Type: `bool`

Default value: `false`

**has\_sysinstr128**

Support for System Instructions that can take 128-bit inputs (FEAT\_SYSINSTR128) values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: `0`

**has\_sysreg128**

Support for 128-bit System Registers (FEAT\_SYSREG128) values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

**has\_tag\_cache\_operations**

Implement tag cache operations DC ZGBVA and DC GBVA (FEAT\_MTETC) values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.6 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

**has\_tcr2**

Implement TCR2\_ELx registers (FEAT\_TCR2) values of this parameter are:- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

**has\_tev**

Support for exception-like overlay management mechanism (FEAT\_TEV) values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if Future Architecture Technologies (FAT) is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

**has\_tidcp1**

Implement Armv8.8 ELO use of implementation defined functionality (FEAT\_TIDCP1) values of this parameter are:- 1, feature is implemented if ARMv8.8 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

**has\_tlb\_conflict\_abort**

Detected inconsistent TLB content generate aborts. Ignored if FEAT\_BBML3 is implemented.

Type: `bool`

Default value: `false`

**has\_tlb\_pa\_caching**

Whether intermediate caching of translation table walks might include NonCoherent caches of previous valid walks. 0, NonCoherent caches might be included. 1, No NonCoherent caches included (FEAT\_nTLBPA).

Type: `bool`

Default value: `false`

**has\_tlbi\_range**

Implement support for TLB Range Maintenance instructions (TLBI RVAE1, etc) from ARMv8.4 (FEAT\_TLBIRANGE).values of this parameter are:- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

**has\_tlbi\_to\_outer\_shareable**

Implement support for TLB Maintenance instructions that extend to the Outer Shareable domain (TLBI VAE1OS, etc) from ARMv8.4 (FEAT\_TLBIOS).values of this parameter are:- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

**has\_tlbi\_ttl**

Implement support for the TTL level hint in by-address TLB Maintenance instructions from ARMv8.4 (FEAT\_TTL).values of this parameter are:- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

**has\_tlbid**

Whether TLBIDomains are implemented in this model. (FEAT\_TLBID)values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.7 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

**has\_tlbiw**

Implement TLBI instruction for stage2 dirty (FEAT\_TLBIW).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.5 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

**has\_translation\_hardenening**

Implement the Translation Hardening Extension (FEAT\_THE)values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

**has\_trbe**

If true, implements the Trace Buffer Extension (FEAT\_TRBE).

Type: `bool`

Default value: `false`

**has\_trbe\_ext**

Implements the Trace Buffer external mode extension (FEAT\_TRBE\_EXT).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

**has\_trc\_ext**

If true, Allow DAP accesses to Trace registers(FEAT\_TRC\_EXT).

Type: `bool`

Default value: `true`

**has\_uinj**

Implement software injection of Undefined Instruction exceptions (FEAT\_UINJ).values of this parameter are:- 1, feature is implemented if ARMv9.6 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

**has\_unaligned\_single\_copy\_atomicity**

Implement support for SCTLR\_ELx.nAA from ARMv8.4, and A64 atomic, exclusive and acquire/release instructions accessing unaligned bytes inside a 16byte window will not generate alignment fault (FEAT\_LSE2).values of this parameter are:- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

**has\_unsupported\_exclusive\_fault**

Report unsupported exclusive access with Unsupported Exclusive fault status (otherwise use external abort).

Type: bool

Default value: true

**has\_v8\_4\_debug\_extension**

Implement ARMv8.4 debug extensions (FEAT\_Debugv8p4).values of this parameter are:- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

**has\_v8\_4\_pmu\_extension**

Implement PMU extension from ARMv8.4 (FEAT\_PMUv3p4).values of this parameter are:- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

**has\_v8\_5\_debug\_over\_power\_down**

Implement ARMv8.5 Debug over powerdown (FEAT\_DoPD).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.5 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

**has\_v8\_5\_pmu\_extension**

Implement PMU extension from ARMv8.5 (FEAT\_PMUv3p5).values of this parameter are:- 1, feature is implemented if ARMv8.5 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

### **has\_v8\_5\_spe\_extension**

Implement SPE extension from ARMv8.5 (FEAT\_SPEv1p1).values of this parameter are:- 1, feature is implemented if ARMv8.5 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

### **has\_v8\_6\_pmu\_events**

Implements PMU events from ARMv8.6.values of this parameter are:- 1, feature is implemented if ARMv8.6 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

### **has\_v8\_7\_fp\_enhancements**

Implements the Floating Point enhancements from Armv8.7 (introduces FPCR.FIZ/AH/NEP, etc. (FEAT\_AFP).)values of this parameter are:- 1, feature is implemented if ARMv8.7 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

### **has\_v8\_7\_pmu\_events**

Implement PMU events from ARMv8.7.values of this parameter are:- 1, feature is implemented if ARMv8.7 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

### **has\_v8\_7\_pmu\_extension**

Implement PMU extension from ARMv8.7 (FEAT\_PMUv3p7).values of this parameter are:- 1, feature is implemented if ARMv8.7 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

### **has\_v8\_7\_spe\_extension**

Implement SPE extension from ARMv8.7 (FEAT\_SPEv1p2)values of this parameter are:- 1, feature is implemented if ARMv8.7 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

#### **has\_v8\_7\_spe\_inverted\_filtering**

Where FEAT\_SPEv1p2 is implemented, whether inverted filtering by events is implemented (represented by PMISDR.FnE).

Type: `bool`

Default value: `true`

#### **has\_v8\_7\_spe\_previous\_branch\_target**

Where FEAT\_SPEv1p2 is implemented, whether the optional branch target feature is implemented (FEAT\_SPE\_PBT).

Type: `bool`

Default value: `true`

#### **has\_v8\_8\_debug\_extension**

Implement ARMv8.8 debug extensions (FEAT\_Debugv8p8) values of this parameter are:- 1, feature is implemented if ARMv8.8 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

#### **has\_v8\_8\_pmu\_events**

Implement PMU events from ARMv8.8 (FEAT\_PMUv3) values of this parameter are:- 1, feature is implemented if ARMv8.8 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

#### **has\_v8\_8\_pmu\_extension**

Implement PMU extension from ARMv8.8 (FEAT\_PMUv3p8) values of this parameter are:- 1, feature is implemented if ARMv8.8 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

#### **has\_v8\_8\_spe\_extension**

Implement SPE extension from ARMv8.8 (FEAT\_SPEv1p3) values of this parameter are:- 1, feature is implemented if ARMv8.8 is enabled.- 2, feature is implemented.

Type: `uint8_t`



Default value: 1

#### **has\_v8\_9\_debug\_extension**

Implement ARMv8.9 debug extensions (FEAT\_Debugv8p9) values of this parameter are:- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

#### **has\_v8\_9\_pc\_sample\_based\_profiling**

Implement PC Sample-based Profiling from ARMv8.9 (FEAT\_PCSRv8p9) values of this parameter are:- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

#### **has\_v8\_9\_pmu\_events**

Implement PMU events from ARMv8.9 (FEAT\_PMUv3) values of this parameter are:- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

#### **has\_v8\_9\_pmu\_extension**

Implement PMU extension from ARMv8.9 (FEAT\_PMUv3p9) values of this parameter are:- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

#### **has\_v8\_9\_spe\_extension**

Implement SPE extension from ARMv8.9 (FEAT\_SPEv1p4) values of this parameter are:- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

#### **has\_v9\_6\_spe\_extension**

Implement FEAT\_SPEv1p5 and FEAT\_SPE\_EXC from ARMv9.6 values of this parameter are:- 1, feature is implemented if ARMv9.6 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

### **has\_virtual\_tag\_check\_on\_load**

When FEAT\_VMTETC is implemented specify whether FEAT\_VMTETCL is implemented. Please note that in order to enable FEAT\_VMTETCL with this parameter, `vmte_support_level` must be set to 1 or 2.

Type: `bool`

Default value: `true`

### **has\_vnchr\_el2**

Implement support for nested virtualization enhancements from ARMv8.4 (FEAT\_NV2).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_wfet\_and\_wfit**

Implements WFE and WFI with Timeout from Armv8.7 (FEAT\_WFxT).values of this parameter are:- 1, feature is implemented if ARMv8.7 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_writebuffer**

Implement write accesses buffering before L1 cache. May affect `ext_abort` behaviour.

Type: `bool`

Default value: `false`

### **has\_xs**

Implements Armv8.7 XS, TLBnXS, DSBnXS instruction (FEAT\_XS).values of this parameter are:- 1, feature is implemented if ARMv8.7 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **hcaptr\_tta\_behaviour**

Behaviour of HCPTR.TTA when there is no CP14 ETM interface. 0, **RAZ/WI**. 1, **RAO/WI**. 2, **stateful**.

Type: `uint8_t`

Default value: 2

**hcr\_el2\_miocnce\_is\_rw**

If true, HCR\_EL2.MIOCNCE is treated as R/W instead of **RAZ/WI**; always set to false in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: false

**hcr\_swio\_res1**

Whether HCR.SWIO and/or HCR\_EL2.SWIO are **RES1**.

Type: bool

Default value: false

**hdbss\_error\_fault\_type**

Type of fault reported for HDBSS errors. 0 = precise exception, 1 = fault logged in HDBSSPROD\_EL2.FSC (FEAT\_HDBSS).

Type: uint8\_t

Default value: 0

**hpfar\_unknown\_when\_ipa\_invalid**

If true, HPFAR\_EL2 is set to 0 when IPA is not valid for stage 2 faults.

Type: bool

Default value: false

**hpfar\_update\_behaviour**

Defines HPFAR\_EL2 update condition. 0: Always updated on faults taken to EL2. 1: Only when IPA is valid. 2: When IPA is valid or unknown.

Type: uint8\_t

Default value: 0

**hsr\_uncond\_cc**

Condition codes reported in HSR as AL if it passes.

Type: bool

Default value: false

**hvbar\_reset\_is\_rvbar**

If true then the reset value of HVBAR is RVBAR, if false the reset value is **UNKNOWN**.

Type: `bool`

Default value: `false`

**icache-hit\_latency**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-log2linelen**

If nonzero, Log2 of the instruction cache line length in bytes (valid values in range 4-8). Otherwise the value of `cache-log2linelen` is used.

Type: `uint8_t`

Default value: `0`

**icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-nprefetch**

Number of next sequential instruction cache lines to prefetch. This is only used when `icache-prefetch_enabled=true`.

Type: `uint32_t`

Default value: `0x1`

**icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

**icache-prefetch\_level**

0 based cache level at which instructions are pre-fetched. This is only used when `icache-prefetch_enabled=true`.

Type: `uint8_t`

Default value: `0`

**icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-read\_bus\_width\_in\_bytes**

L1 I-Cache read bus width in bytes used to calculate per-access timing annotations.

Type: `uint32_t`

Default value: `0x8`

**icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-size**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: 0x8000

### **icache-state\_modelled**

Set whether I-cache has stateful implementation.

Type: bool

Default value: false

### **icache-ways**

L1 I-Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 2

### **id\_spec\_fpacc\_raz**

If true, implementation opts not to disclose the speculative use of pointers processed by a PAC authentication failure by having value 0 for Spec\_FPACC bits of ID\_AA64MMFR3\_EL1 register.

Type: bool

Default value: false

### **idte3\_support\_level**

Support for trapping ID register accesses to EL3(FEAT\_IDTE3): 0 - Not implemented. 1 - Implemented.

Type: uint8\_t

Default value: 0

### **iesb\_use\_pre\_cse\_ctx**

If true, read SCTLR\_ELx.IESB and SCR\_EL3.{NMEA,EA} before synchronizing registers. The pre-synchronization values will be used to determine whether to take an implicit ESB.

Type: bool

Default value: false

### **ignore\_DBGPRCR\_CWRR**

Ignore writes to the deprecated DBGPRCR.CWRR bit.

Type: bool

Default value: false

**ignore\_access\_flag\_update\_by\_CMOs**

If true, CMOs(cache maintenance operations) neither update the access flag nor generate access flag fault.

Type: `bool`

Default value: `false`

**ignore\_access\_flag\_update\_by\_at\_ops**

If true, AT operations do not update access flag.

Type: `bool`

Default value: `false`

**ignore\_data\_abt\_on\_af\_update\_by\_at\_ops**

If true, Data abort generated on AF update by AT operations are ignored. This parameter is only valid if `ignore_access_flag_update_by_at_ops` is false.

Type: `bool`

Default value: `true`

**ignore\_large\_address\_top\_bits\_in\_page\_walk**

Whether page table bits [15:12] are ignored if `PA_SIZE < 52` and output address is configured `< 52` with large page.

Type: `bool`

Default value: `false`

**ignore\_tag\_check\_dcc\_load\_store\_in\_ma\_mode\_when\_tco\_is\_disabled**

Constrained unpredictable behavior for reads/writes to external debug interface DTR regs in memory access mode when `PSTATE.TCO` is 0. If true, tag check is ignored else, tag check is performed if required.

Type: `bool`

Default value: `false`

**ignore\_traps\_to\_dcc\_regs\_in\_debug**

Whether traps get ignored for the following registers in debug state:AArch64: MDCCSR\_EL0, OSDTREX\_EL1, OSDTRTX\_EL1, MDCCINT\_EL1.AArch32: DBGDSCRint, DBGDIDR, DBGDSAR, DBGDRAR,, DBGDTRTXext, DBGDCCINT.

Type: `bool`

Default value: `false`

**illegal\_state\_exception\_priority**

IMPDEF priority of Illegal State Exception. 0: After breakpoint exceptions 1: Before Instruction Abort.

Type: `uint8_t`

Default value: 0

**imp\_def\_functionality\_behaviour**

Behaviour of **IMPLEMENTATION DEFINED** registers and system instructions. 0, UNDEF. 1, **RAZ/WI**.

Type: `uint8_t`

Default value: 0

**impdef\_regs\_and\_unpred\_from\_implementation**

Configure implementation defined registers and unpredictable behaviour to match the specified implementation. Requires a license for the selected implementation model. User has to provide the default values for the published or configurable parameters through commandline arguments. Use `ARM_Cortex-A<num>` or `ARM_<codename>` for licensed pre-release cores.

Type: `string`

Default value: `N/A`

**impdef\_sysreg\_json**

Configure mask/reset for impdef registers in a JSON format. a string (max 1024 chars) or a filename, starting with '@' sign. is a list of objects, with following attributes:.

Type: `string`

Default value: `[]`

**independent\_cache\_control\_traps**

Implement Independent Cache Control traps from ARMv8.5. 0, No support. 1, Supported but not for tlb maintenance instructions. 2, Full support. (FEAT\_EVT).

Type: `uint8_t`

Default value: 0

**inner\_cache\_boundary**

CLIDR.ICB, cache level boundary between inner and outer shareable domains.

Type: `uint8_t`

Default value: 0



**insert\_iesb\_before\_exception**

If true then inserts an IESB before taking with Exception otherwise has no effect and IESB is taken after PState is changed due to the Exception.

Type: `bool`

Default value: `false`

**instruction\_tlb\_size**

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

Type: `uint32_t`

Default value: `0x0`

**internal\_vgic**

Instantiate VGIC peripheral in this processor.

Type: `bool`

Default value: `false`

**invalidate\_code\_cache\_on\_icache\_cmo**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: `uint8_t`

Default value: `0`

**irt\_fetch\_fault\_on\_data\_check\_report\_type**

Behaviour for IRT fetch faults on data permission checks (can only happen for non-standard data accesses): 0, report as data abort. 1, report as instruction abort.

Type: `bool`

Default value: `false`

**is\_debug\_state\_pmu\_snapshot\_allowed**

If true, PMU snapshot is allowed in debug state.

Type: `bool`

Default value: `true`

**is\_first\_pcsr\_sample\_ignored**

If true, First read of PMPCSR register after reset returns `0xFFFFFFFF`.

Type: `bool`

Default value: `false`

### **`is_mt_res0`**

If ARMv8.6 is not implemented, and PMUv3 is implemented, this parameter controls whether PMEVTYPER<n>.MT bit is **RES0** or RW. For other implementations, this parameter has no effect.

Type: `bool`

Default value: `false`

### **`is_ras_irq_edge_triggered`**

If true, ras interrupt is edge-triggered. Otherwise, it's level-triggered.

Type: `bool`

Default value: `true`

### **`is_serror_edge_triggered`**

If true, SError is edge-triggered. Otherwise, it's level-triggered.

Type: `bool`

Default value: `true`

### **`is_tagged_nsh_treated_as_tagged`**

Whether a tagged NonShared memory attribute is treated as tagged or not.

Type: `bool`

Default value: `true`

### **`is_uniprocessor`**

Value for the U bit in MPIDR. true disables L1 cache coherency protocols.

Type: `bool`

Default value: `false`

### **`isb_is_branch`**

If true, ISB is considered an immediate branch. This allows to count ISB as a branch in BRBE.

Type: `bool`

Default value: `false`

**ish\_is\_osh**

Whether Innershareable is same as OuterShareable.

Type: `bool`

Default value: `false`

**itd\_conditional\_instructions\_are\_32bit**

When SCTLR\_ELx.ITD=1, an IT instruction plus a T16 instruction are considered a single 32bit conditional instruction.

Type: `bool`

Default value: `false`

**jidr\_is\_undef\_at\_el0**

If true, JIDR register access is UNDEF at EL0.

Type: `bool`

Default value: `false`

**jmcr\_is\_undef\_at\_el0**

If true, JMCR register access is UNDEF at EL0.

Type: `bool`

Default value: `false`

**joscr\_is\_undef\_at\_el0**

If true, JOSCR register access is UNDEF at EL0.

Type: `bool`

Default value: `false`

**l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l2cache-read_bus_width_in_bytes`**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: `uint32_t`

Default value: `0x8`

### **`l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

**l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 16

**l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x8

**l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**l3cache-has\_mpam**

L3 Cache has MPAM support.

Type: `bool`

Default value: `false`

**l3cache-hit\_latency**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-maintenance\_latency**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-miss\_latency**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-mpamf.arch\_major\_ver**

L3 Cache MPAMF\_AIDR architecture major version.

Type: `uint8_t`

Default value: `0`

**l3cache-mpamf.arch\_minor\_ver**

L3 Cache MPAMF\_AIDR architecture minor version.

Type: `uint8_t`

Default value: `0`

**l3cache-mpamf.bwa\_width\_ns**

L3 Cache width of MPAM bandwidth allocation fields for non-secure accesses.

Type: `uint8_t`

Default value: 16

### **`l3cache-mpamf.bwa_width_s`**

L3 Cache width of MPAM bandwidth allocation fields for secure accesses.

Type: `uint8_t`

Default value: 16

### **`l3cache-mpamf.cmax_width_ns`**

L3 Cache, number of fractional bits used to calculate the maximum fraction of the MPAM cache capacity for the Non-Secure PARTID. Only the register interface is implemented - the control is NOT FUNCTIONAL.

Type: `uint8_t`

Default value: 0

### **`l3cache-mpamf.cmax_width_rl`**

L3 Cache, number of fractional bits used to calculate the maximum fraction of the MPAM cache capacity for the Realm PARTID. Only the register interface is implemented - the control is NOT FUNCTIONAL.

Type: `uint8_t`

Default value: 0

### **`l3cache-mpamf.cmax_width_rt`**

L3 Cache, number of fractional bits used to calculate the maximum fraction of the MPAM cache capacity for the Root PARTID. Only the register interface is implemented - the control is NOT FUNCTIONAL.

Type: `uint8_t`

Default value: 0

### **`l3cache-mpamf.cmax_width_s`**

L3 Cache, number of fractional bits used to calculate the maximum fraction of the MPAM cache capacity for the Secure PARTID. Only the register interface is implemented - the control is NOT FUNCTIONAL.

Type: `uint8_t`

Default value: 0

**l3cache-mpamf.cmin\_width\_ns**

L3 Cache, number of fractional bits used to calculate the minimum fraction of the MPAM cache capacity for the Non-Secure PARTID. Only the register interface is implemented - the control is NOT FUNCTIONAL.

Type: uint8\_t

Default value: 0

**l3cache-mpamf.cmin\_width\_rl**

L3 Cache, number of fractional bits used to calculate the minimum fraction of the MPAM cache capacity for the Realm PARTID. Only the register interface is implemented - the control is NOT FUNCTIONAL.

Type: uint8\_t

Default value: 0

**l3cache-mpamf.cmin\_width\_rt**

L3 Cache, number of fractional bits used to calculate the minimum fraction of the MPAM cache capacity for the Root PARTID. Only the register interface is implemented - the control is NOT FUNCTIONAL.

Type: uint8\_t

Default value: 0

**l3cache-mpamf.cmin\_width\_s**

L3 Cache, number of fractional bits used to calculate the minimum fraction of the MPAM cache capacity for the Secure PARTID. Only the register interface is implemented - the control is NOT FUNCTIONAL.

Type: uint8\_t

Default value: 0

**l3cache-mpamf.cpbm\_width\_ns**

L3 Cache, width of MPAM cache portion bitmap for non-secure accesses. If 0, the feature is not implemented, and all ways are available.

Type: uint8\_t

Default value: 0

**l3cache-mpamf.cpbm\_width\_rl**

L3 Cache, width of MPAM cache portion bitmap for realm accesses. If 0, the feature is not implemented, and all ways are available.



Type: `uint8_t`

Default value: 0

### **`l3cache-mpamf.cpbm_width_rt`**

L3 Cache, width of MPAM cache portion bitmap for root accesses. If 0, the feature is not implemented, and all ways are available.

Type: `uint8_t`

Default value: 0

### **`l3cache-mpamf.cpbm_width_s`**

L3 Cache, width of MPAM cache portion bitmap for secure accesses. If 0, the feature is not implemented, and all ways are available.

Type: `uint8_t`

Default value: 0

### **`l3cache-mpamf.csu_num_mon_ns`**

L3 Cache number of MPAM cache storage usage monitors for non-secure accesses.

Type: `uint16_t`

Default value: 0x0

### **`l3cache-mpamf.csu_num_mon_rl`**

L3 Cache number of MPAM cache storage usage monitors for realm accesses.

Type: `uint16_t`

Default value: 0x0

### **`l3cache-mpamf.csu_num_mon_rt`**

L3 Cache number of MPAM cache storage usage monitors for root accesses.

Type: `uint16_t`

Default value: 0x0

### **`l3cache-mpamf.csu_num_mon_s`**

L3 Cache number of MPAM cache storage usage monitors for secure accesses.

Type: `uint16_t`

Default value: 0x0

**l3cache-mpamf.esr\_mask**

L3 Cache MPAMF\_ESR mask value.

Type: `uint32_t`

Default value: `0xffffffff`

**l3cache-mpamf.has\_esr**

L3 Cache's MPAMF\_ESR, MPAMF\_ECR, and MPAM error handling implemented.

Type: `bool`

Default value: `false`

**l3cache-mpamf.has\_extd\_esr**

L3 Cache's MPAMF\_ESR is 64-bits.

Type: `bool`

Default value: `false`

**l3cache-mpamf.has\_impl\_idr**

L3 Cache's MPAMF\_IMPL\_IDR is present.

Type: `bool`

Default value: `false`

**l3cache-mpamf.has\_mbwu\_long\_counter**

L3 Cache has long MBWU counter and capture registers.

Type: `bool`

Default value: `false`

**l3cache-mpamf.has\_mpamfidr\_ext**

MPAMF\_IDR.EXT support.

Type: `bool`

Default value: `false`

**l3cache-mpamf.has\_partid\_nrw**

Narrowing part ID register is present. This is global rather than per-instance.

Type: `bool`

Default value: `false`

**l3cache-mpamf.has\_priority\_partitioning**

The selected resource has priority partitioning described in MPAMF\_PRI\_IDR.

Type: `bool`

Default value: `false`

**l3cache-mpamf.has\_prod\_id**

L3 Cache MPAMF\_IIDR product ID supported.

Type: `uint16_t`

Default value: `0x0`

**l3cache-mpamf.has\_prod\_rev**

L3 Cache MPAMF\_IIDR product REVISION supported.

Type: `uint8_t`

Default value: `0`

**l3cache-mpamf.has\_prod\_var**

L3 Cache MPAMF\_IIDR product VARIANT supported.

Type: `uint8_t`

Default value: `0`

**l3cache-mpamf.has\_prop\_ns**

Enable memory bandwidth proportional stride control for non-secure accesses. Only the register interface is implemented - the control is NOT FUNCTIONAL.

Type: `bool`

Default value: `false`

**l3cache-mpamf.has\_prop\_s**

Enable memory bandwidth proportional stride control for secure accesses. Only the register interface is implemented - the control is NOT FUNCTIONAL.

Type: `bool`

Default value: `false`

**l3cache-mpamf.has\_ris**

L3 Cache has resource instance selection support.

Type: `bool`

Default value: `false`

### **l3cache-mpamf.max\_partid\_ns**

L3 Cache Maximum value of non-secure PARTID supported.

Type: `uint16_t`

Default value: `0xffff`

### **l3cache-mpamf.max\_partid\_rl**

L3 Cache Maximum value of realm PARTID supported for RME implementations.

Type: `uint16_t`

Default value: `0xffff`

### **l3cache-mpamf.max\_partid\_rt**

L3 Cache Maximum value of root PARTID supported for RME implementations.

Type: `uint16_t`

Default value: `0xffff`

### **l3cache-mpamf.max\_partid\_s**

L3 Cache Maximum value of secure PARTID supported.

Type: `uint16_t`

Default value: `0xffff`

### **l3cache-mpamf.max\_pmg\_ns**

L3 Cache Maximum value of non-secure PMG supported.

Type: `uint8_t`

Default value: 255

### **l3cache-mpamf.max\_pmg\_rl**

L3 Cache Maximum value of realm PMG supported for RME implementations.

Type: `uint8_t`

Default value: 255

### **l3cache-mpamf.max\_pmg\_rt**

L3 Cache Maximum value of root PMG supported for RME implementations.

Type: `uint8_t`

Default value: 255

### **l3cache-mpamf.max\_pmg\_s**

L3 Cache Maximum value of secure PMG supported.

Type: `uint8_t`

Default value: 255

### **l3cache-mpamf.mbwu\_long\_counter\_width**

L3 Cache long MBWU counter width. 0: 63 bits, 1: 44 bits.

Type: `uint8_t`

Default value: 0

### **l3cache-mpamf.no\_impl\_msmon**

L3 Cache's MPAMF\_IMPL\_IDR does not describe resource monitors.

Type: `bool`

Default value: `false`

### **l3cache-mpamf.no\_impl\_part**

L3 Cache's MPAMF\_IMPL\_IDR does not describe resource partitioning controls.

Type: `bool`

Default value: `false`

### **l3cache-mpamf.ris\_max**

L3 Cache's largest resource instance selector value defined.

Type: `uint8_t`

Default value: 0

### **l3cache-mpamf\_base**

L3 Cache memory mapped MPAM registers base address.

Type: `uint64_t`

Default value: 0x0

**l3cache-read\_access\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size l3cache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**l3cache-read\_bus\_width\_in\_bytes**

L3 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x8

**l3cache-read\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. l3cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**l3cache-size**

L3 Cache size in bytes.

Type: uint32\_t

Default value: 0x0

**l3cache-snoop\_data\_transfer\_latency**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**l3cache-snoop\_issue\_latency**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **l3cache-ways**

L3 Cache number of ways (sets are implicit from size).

Type: `uint16_t`

Default value: 16

### **l3cache-write\_access\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size `l3cache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **l3cache-write\_bus\_width\_in\_bytes**

L3 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: `uint32_t`

Default value: 0x8

### **l3cache-write\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l3cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **legacy\_combining\_exc\_catch\_trace**

Whether exception catch is traced as part of exception entry/exit in same cycle.

Type: `bool`

Default value: `true`

### **limit\_ete\_revision\_without\_rme**

If true, limit the ETE revision to ETEv1p1 when FEAT\_RME is not implemented.

Type: `bool`

Default value: `false`

**log2\_trace\_buffer\_alignment**

Log2 of trace buffer alignment constraint for output buffer (0->1B ... 11->2Kib).

Type: `uint8_t`

Default value: 0

**ls64\_ignore\_s1\_unpred\_memattr\_transformation**

If true, stage 1 unpredictable memory attribute transformations are ignored for FEAT\_LS64 single-copy atomic 64-byte load/store instructions' (FEAT\_LS64, FEAT\_LS64\_V, FEAT\_LS64\_ACCDATA).

Type: `bool`

Default value: `false`

**ls64\_memtype\_check\_use\_combined\_memattr**

FEAT\_LS64 single-copy atomic 64-byte load/store instructions' 0 : memory attributes check is performed at each enabled stage of translation, 1 : memory attributes check is done on the combined memory attributes only. 2. memory attributes check is done on the combined memory attributes with Stage1 and Stage2 fault get evaluated to check on which stage fault should be reported.

Type: `uint8_t`

Default value: 0

**ls64wb\_memtype\_check\_allow\_any\_cacheable\_memattr**

If true, when FEAT\_LS64WB is implemented, any cacheable memory access performed by LD/ST64B instructions is 64-byte, single-copy atomic.

Type: `bool`

Default value: `false`

**mask\_trbtrg\_res0**

If true, and TRBIDR\_EL1.Align>0, treat TRBTRG\_EL1[TRBIDR\_EL1.Align-1:0] as **RES0** for writes.

Type: `bool`

Default value: `false`

**max\_32bit\_el**

Maximum exception level supporting AArch32 modes. -1: No Support for A32 at any EL, x:[0:3] - All the levels below supplied ELx supports A32.

Type: `int8_t`

Default value: 3



**mdrar\_el1\_res0**

MDRAR\_EL1 is **RES0**.

Type: `bool`

Default value: `false`

**mdselr\_le\_16\_bps\_wps\_behaviour**

Behaviour of MDSELR\_EL1 and related traps/enables if fewer than 16 watchpoints and fewer than 16 breakpoints are implemented: 0 - MDSELR\_EL1 is stateful; 1 - MDSELR\_EL1, EBWE, FGTS are **RAZ/WI**, traps and enables do not apply; 2 - MDSELR\_EL1, EBWE, FGTS with checked traps.

Type: `uint8_t`

Default value: 0

**mec\_support\_level**

0 -> Memory Encryption Contexts not implemented, 1 -> LEGACY\_TZ\_EN mode i.e. MEC register fields are stateful but only supports secure/non-secure states, 2 -> Memory Encryption Contexts fully implemented (FEAT\_MEC).

Type: `uint8_t`

Default value: 0

**memory.acp.AxCACHE\_mask**

Used with `memory.acp.AxCACHE_pattern` to define which memory types the ACP port accepts. All transactions which do not satisfy  $(\text{AxCACHE} \& \text{mask}) == \text{pattern}$  will abort.

Type: `uint8_t`

Default value: 0

**memory.acp.AxCACHE\_pattern**

Used with `memory.acp.AxCACHE_mask` to define which memory types the ACP port accepts. All transactions which do not satisfy  $(\text{AxCACHE} \& \text{mask}) == \text{pattern}$  will abort.

Type: `uint8_t`

Default value: 0

**memory.l2\_cache.is\_inner\_cacheable**

L2 cache obeys inner cacheable attributes (rather than outer cacheable attributes).

Type: `bool`

Default value: `true`

**memory.l2\_cache.is\_inner\_shareable**

L2 cache obeys inner shareable attributes (rather than outer shareable attributes).

Type: `bool`

Default value: `true`

**memory\_tagging\_support\_level**

Specify the memory tagging extension support level: 0, not implemented. 1, instructions and registers only are implemented (FEAT\_MTE). 2, implemented (FEAT\_MTE2). 3, implemented with asymmetric handling of exceptions (FEAT\_MTE3). 4, implemented (FEAT\_MTE4).

Type: `uint8_t`

Default value: 0

**mixed\_endian**

Implement support for mixed endianness. 0, not supported. 1, supported at all exception levels. 2, supported at ELO only. Unsupported in the presence of Future Architecture Technologies (FAT).

Type: `uint8_t`

Default value: 1

**mops\_cpy\_block\_size**

Block size used for memcpy memory accesses.

Type: `uint8_t`

Default value: 64

**mops\_cpy\_default\_dir**

Default direction for non-overlapping memcpy operations: 0, forwards. 1, backwards.

Type: `uint8_t`

Default value: 0

**mops\_cpy\_handle\_async\_exceptions**

Handle any pending async exceptions after copying a block of data, instead of waiting until instruction end.

Type: `bool`

Default value: `false`

**mops\_cpy\_post\_size**

Percentage of data copied in memcpy 'E' instructions.

Type: `uint8_t`

Default value: 10

**mops\_cpy\_pre\_size**

Percentage of data copied in memcpy 'P' instructions.

Type: `uint8_t`

Default value: 10

**mops\_cpy\_pre\_size\_threshold**

Size threshold in Bytes for CPYP instructions.

Type: `uint32_t`

Default value: 0x0

**mops\_cpy\_single\_access**

Execute memcpy as a single read and single write access.

Type: `bool`

Default value: `false`

**mops\_cpy\_write\_abort\_before\_read**

Report the data aborts and watchpoint of the write accesses, before those of the read accesses.

Type: `bool`

Default value: `false`

**mops\_cpy\_zero\_size\_can\_fault**

Fault because of mismatched implementation option when the operation is of size 0.

Type: `bool`

Default value: `true`

**mops\_exec\_order\_can\_fault**

Enable exception on the Main/Epilogue instruction when executed after a mismatched Prologue/Main in a CPY/SET sequence, or after another random instruction.

Type: `bool`

Default value: `false`

**mops\_inst\_cpy\_zero\_size\_can\_fault**

Fault because of mismatched implementation option when `inst_cpy_size` is 0.

Type: `bool`

Default value: `true`

**mops\_inst\_set\_zero\_size\_can\_fault**

Fault because of mismatched implementation option when `inst_set_size` is 0.

Type: `bool`

Default value: `true`

**mops\_mismatched\_page\_crossing\_access\_unpred**

Constrained unpredictable behaviour for FEAT\_MOPS instructions when crossing page boundary with different memory types, 0 : Memory block access uses the attributes of it's own address block  
1: Alignment Fault.

Type: `uint8_t`

Default value: 0

**mops\_mmu\_abort\_far\_aligned**

If true, in case of an MMU abort on a MOPS instruction, report FAR aligned to current translation granule.

Type: `bool`

Default value: `false`

**mops\_set\_block\_size**

Block size used for `memset` memory accesses.

Type: `uint8_t`

Default value: 64

**mops\_set\_handle\_async\_exceptions**

Handle any pending async exceptions after setting a block of data, instead of waiting until instruction end.

Type: `bool`

Default value: `false`

**mops\_set\_post\_size**

Percentage of data copied in memset 'E' instructions.

Type: `uint8_t`

Default value: 10

**mops\_set\_pre\_size**

Percentage of data copied in memset 'P' instructions.

Type: `uint8_t`

Default value: 10

**mops\_set\_single\_access**

Execute memset as a single read and single write access.

Type: `bool`

Default value: `false`

**mops\_set\_zero\_size\_can\_fault**

Fault because of mismatched implementation option when the operation is of size 0.

Type: `bool`

Default value: `true`

**mops\_setg\_unaligned\_does\_mismatch\_fault**

If true, in case of unaligned SETGM / SETGE, raise a mismatched memset exception because of impdef reasons, instead of alignment fault.

Type: `bool`

Default value: `false`

**mops\_wp\_far\_behaviour**

Set option for address stored in FAR/EDWARD after watchpoints hit by MOPS instructions

- FAR recorded matches lowest watchpointed address accessed by the instruction
- FAR recorded matches lowest address accessed by the instruction within same translation granule as watchpointed address
- FAR recorded matches highest watchpointed address accessed by the instruction that triggered the watchpoint.

Type: `uint8_t`

Default value: 0

**mpam\_at\_ops\_use\_target\_el**

If true, AT instruction table fetches use the owning translation regime's EL for MPAM PARTID/PMG selection.

Type: `bool`

Default value: `false`

**mpam\_bw\_bwa\_wd**

MPAM MPAMBWIDR\_EL1.BWA\_WD: The number of implemented bits in the bandwidth allocation fields {MPAMBWn\_ELx, MPAMBWSM\_EL1}.MAX and MPAMBWCAP\_EL2.CAP.

Type: `uint8_t`

Default value: 1

**mpam\_bw\_has\_hw\_scale**

MPAM Whether has hardware support for auto-scaling of {MPAMBWn\_ELx, MPAMBWSM\_EL1}.MAX and MPAMBWCAP\_EL2.CAP limits.

Type: `bool`

Default value: `false`

**mpam\_bw\_max\_lim**

MPAM the implemented maximum-bandwidth limit partitioning behaviors:- 0, Both soft limit and hard limit behaviors are implemented.- 1, Soft limit behavior is implemented.- 2, Hard limit behavior is implemented.

Type: `uint8_t`

Default value: 0

**mpam\_bw\_us\_frac**

MPAM MPAMBWIDR\_EL1.US\_FRAC: The fractional part of the window width in microseconds.

Type: `uint8_t`

Default value: 0

**mpam\_bw\_us\_int**

MPAM MPAMBWIDR\_EL1.US\_INT: The integer part of the window width in microseconds.

Type: `uint32_t`

Default value: 0x0

**mpam\_force\_ns\_rao**

Whether MPAM3\_EL3.FORCE\_NS bit is **RAO/WI**.

Type: `bool`

Default value: `false`

**mpam\_frac**

MPAM fractional revision number in ID\_AA64PFR1\_EL1.MPAM\_frac field. Combines with `has_mpam` to give the mpam version `mpam = false, mpam_frac = 0` -> Not implemented `mpam = false, mpam_frac = 1` -> FEAT\_MPAMvOp1 `mpam = true, mpam_frac = 0` -> FEAT\_MPAMv1p0 `mpam = true, mpam_frac = 1` -> FEAT\_MPAMv1p1.

Type: `uint8_t`

Default value: `0`

**mpam\_has\_altsp**

DEPRECATED: MPAMIDR\_EL1.HAS\_ALTSP is required when FEAT\_RME is implemented.

Type: `bool`

Default value: `false`

**mpam\_has\_bw\_ctrl**

MPAM Whether MPAMIDR\_EL1.HAS\_BW\_CTRL bit is set or clear. values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.3 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: `0`

**mpam\_has\_hcr**

MPAM Whether MPAMIDR\_EL1 HAS\_HCR bit is set or clear.

Type: `bool`

Default value: `false`

**mpam\_max\_partid**

MPAM Maximum PARTID Supported.

Type: `uint16_t`

Default value: `0xffff`

**mpam\_max\_pmg**

MPAM Maximum PMG Supported.

Type: `uint16_t`

Default value: `0xffff`

**mpam\_max\_vpmr**

MPAM Maximum VPMR Supported.

Type: `uint8_t`

Default value: `0`

**mpam\_truncate\_out\_of\_range\_virtid**

If true then truncates an out-of-range virtual identifier to least significant 12 bits. If false then out-of-range virtual identifier is replaced by the default virtual identifier.

Type: `bool`

Default value: `false`

**mpamidr\_has\_force\_ns**

Whether MPAMIDR\_EL1.HAS\_FORCE\_NS bit is set or clear. values of this parameter are:-  
0, feature is not enabled.- 1, feature is implemented if ARMv8.6 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: `0`

**mpamidr\_has\_sdeflt**

Whether MPAMIDR\_EL1.HAS\_SDEFLT bit is set or clear. values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.6 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: `0`

**mpamidr\_has\_tidr**

Whether MPAMIDR\_EL1.HAS\_TIDR bit is set or clear. values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.6 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: `0`



**mpidr\_layout**

Layout of MPIDR. 0 AFF0 is CPUID, 1 AFF1 is CPUID.

Type: `uint8_t`

Default value: 0

**mpmm\_accumulator\_multiplier**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator. value of  $n$  means the accumulator will use  $(n * \text{accumulator value})$  to calculate the mpmm threshold (MPMM). is provided as a fast model workaround to handle cases where execution of quantum in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

Type: `uint8_t`

Default value: 1

**mpmm\_config**

MPMMTUNE register value. The JSON schema is : . The value given for threshold value is just an indication, not specific to any core. This parameter is used only when `has_mpmm` is set.

Type: `string`

Default value: N/A

**mte\_ctrl\_bits\_stateful\_level**

If `memory_tagging_support_level == 1`, specify the MTE level that has control bits stateful in system registers.

Type: `uint8_t`

Default value: 0

**mte\_report\_which\_failed\_address**

Set to `<OPT>`, `<MOPS_OPT>` Applicable only for MTE synchronous check. `OPT` defines the range for the failing address to report and `MOPS_OPT` defines the choice within that range for MOPS operations only. Non-MOPS operations report the first address in the range defined by `OPT`. `OPT` is set to "first" or "last". If "first" then report an address from the intersection of the first failed MTE granule and the transaction's range. If "last" then report an address from the intersection of the last failed MTE granule and the transaction's range. `MOPS_OPT` is set to "mops\_first\_failing\_address\_in\_range" or "mops\_random\_address\_in\_range". If "mops\_first\_failing\_address\_in\_range" then report the first failing address in the range defined by `OPT`. If "mops\_random\_address\_in\_range" then report a random address within the range defined by `OPT`.

Type: `string`

Default value: "first, mops\_first\_failing\_address\_in\_range"

### **mte\_tminline**

Value of CTR\_EL0.TminLine for reading purpose only. A value configured using this does not indicate the presence of separate tag cache. 0, TminLine evaluated from smallest data cache line.

Type: uint8\_t

Default value: 0

### **mte\_unpred\_canonical\_s2\_unsupported**

When FEAT\_MTE\_CANONICAL\_TAGS is implemented, determines whether a region is canonically tagged (true) or untagged (false) when stage 1 is tagged but the combined attributes do not allow allocation tagging.

Type: bool

Default value: false

### **mvbar\_reset\_is\_rvbar**

If true then the reset value of MVBAR is RVBAR, if false the reset value is **UNKNOWN**.

Type: bool

Default value: true

### **non\_secure\_vgic\_alias\_when\_ns\_only**

If ! has\_el3 and only non-secure side exists, then the normal position of the VGIC is a secure alias. If this parameter is non-zero then in addition a non-secure alias of the VGIC will be placed at this position (aligned to 32 KB).

Type: uint64\_t

Default value: 0x0

### **num\_loregion\_descriptors**

Number of Limited Ordering Region descriptors implemented (if ARM v8.1 extensions are implemented) (FEAT\_LOR).

Type: uint8\_t

Default value: 0

### **num\_loregions**

Number of Limited Ordering Regions implemented excluding background region (if ARM v8.1 extensions are implemented) (FEAT\_LOR).

Type: uint8\_t

Default value: 0

### **number\_of\_abl\_breakpoints**

if FEAT\_ABL is implemented, Number of address matching breakpoints that support address linking.

Type: uint8\_t

Default value: 0

### **number\_of\_error\_records**

Cores Number of Error records supported for RAS.

Type: uint16\_t

Default value: 0x0

### **nv\_frac\_support\_level**

Support for a subset of FEAT\_NV and FEAT\_NV2 behaviours: 0 - Not implemented. 1 - Implemented. 2 - Implemented with FEAT\_NV2p1. 3 - Implemented with FEAT\_NV3, OPTIONAL from Armv9.6.

Type: uint8\_t

Default value: 0

### **optimal\_alignment\_size**

Alignment boundary which does not incur additional performance penalty from ARMv8.5.- 1, architectural misalignment is used to set PMU event LDST\_ALIGN\_LAT and SPE event E[11]- 2, access crossing 4 byte boundary is used to set PMU event LDST\_ALIGN\_LAT and SPE event E[11]- 3, access crossing 8 byte boundary is used to set PMU event LDST\_ALIGN\_LAT and SPE event E[11].- 12, access crossing 4 Kbyte boundary is used to set PMU event LDST\_ALIGN\_LAT and SPE event E[11].

Type: uint8\_t

Default value: 1

### **output\_attributes**

User-defined transform to be applied to bus attributes like ManagerID, ExtendedID or UserFlags. Currently, only works for MPAM Attributes encoding into bus attributes.

Type: string

Default value: "ExtendedID[62:55]=MPAM\_PMG, ExtendedID[54:39]=MPAM\_PARTID, ExtendedID[38]=MPAM\_NS"

**pacm\_support\_level**

Implements PSTATE.PACM from ARMv9.5. 0: Not supported, 1: Trivial implementation when FEAT\_PAuth\_LR and FEAT\_PACIMP are supported, 2: Full implementation when FEAT\_PAuth\_LR is supported.

Type: `uint8_t`

Default value: 2

**page\_based\_hardware\_attributes**

Implement the page based hardware attributes from ARMv8.2. This parameter indicates which page table bits are available for hardware, where bits[3:0] correspond to PTE[62:59] and to TCR\_ELx.HWUnyyy (FEAT\_HPDS2).

Type: `uint8_t`

Default value: 0

**pan\_applies\_before\_oa\_space\_checks**

Apply effects of PAN on S1 indirect permissions before OA space security checks.

Type: `bool`

Default value: `true`

**pan\_removes\_priv\_rw\_if\_unpriv\_resvd\_value**

For regimes EL1&0/EL2&0, PSTATE.PAN = 1, whether PrivRW is removed when S1UnprivBasePerm is reserved.

Type: `bool`

Default value: `true`

**pan\_stage1\_in\_realm\_el2\_0\_is\_unx**

If FEAT\_PAN3 is implemented, whether stage1 translation in the Realm EL2&0 regime that resolves to a NS address is treated as unprivileged execute-never.

Type: `bool`

Default value: `false`

**par\_ns\_set\_unknown\_bit**

Whether NS bit of PAR is set/clear when executing AT to perform non-secure regime translation. When true, NS is set to 1 else 0.

Type: `bool`

Default value: `true`

**par\_nse\_set\_unknown\_bit**

Whether NSE bit of PAR is set/clear when executing AT operation on secure, non-secure or realm translation regime. When true, NSE is set to 1 else 0.

Type: `bool`

Default value: `false`

**permission\_overlay\_s1\_support\_level**

Support for Stage 1 Permission Overlay: 0 - None. 1 - Support FEAT\_S1POE. 2 - Support FEAT\_S1POE2.

Type: `uint8_t`

Default value: 0

**pfar\_is\_valid**

**IMPLEMENTATION DEFINED** choice to configure ESR\_ELx.PFV: whether PFAR\_ELx is valid or **UNKNOWN** when ESR\_ELx.PFV is not forced to be 0.

Type: `bool`

Default value: `true`

**pfr1\_csv2\_frac**

Fractional revision number ID\_AA64PFR1\_EL1.CSV2\_frac when ID\_AA64PFR0\_EL1.CSV==1 for CSV2 extension (FEAT\_CSV2\_1p1, FEAT\_CSV2\_1p2).

Type: `uint8_t`

Default value: 0

**plb\_dpote\_enabled**

Enable PLB for POE2 DPOT lookups. NOTE: setting this parameter to false will significantly degrade model performance.

Type: `bool`

Default value: `true`

**plb\_irt\_enabled**

Enable PLB for POE2 IRT lookups. NOTE: setting this parameter to false will significantly degrade model performance.

Type: `bool`

Default value: `true`

**plb\_ttt\_enabled**

Enable PLB for POE2 TTT lookups. NOTE: setting this parameter to false will significantly degrade model performance.

Type: `bool`

Default value: `true`

**plbi\_invalid\_xt**

If true, PLBI instructions with invalid Xt are treated as UNDEF.

Type: `bool`

Default value: `false`

**pmb\_idr\_external\_abort**

Describes how the PE manages External aborts on writes made by the Statistical Profiling Extension to the Profiling Buffer. 0, External abort is reported to SPE, From Armv8.8 and Armv9.3, the value 0 is not permitted. 1, External abort is ignored. 2, The External abort generates an SError and the error is not reported to SPE.

Type: `uint8_t`

Default value: `0`

**pmb\_idr\_flag\_updates**

Defines whether the address translation performed by the Profiling Buffer manages the Access Flag and dirty state.

Type: `bool`

Default value: `true`

**pmbsr\_dl\_razwi**

Whether PMBSR\_ELx.DL is **RAZ/WI** or behaves as specified, indicating partial loss of a record due to a buffer management event or external abort.

Type: `bool`

Default value: `false`

**pmbsr\_ea\_razwi**

Whether PMBSR\_ELx.EA is **RAZ/WI** or set as the result of an external abort.

Type: `bool`

Default value: `false`

**`pmc_r_disable_events_export`**

If true, export for PMU events is disabled. This configures PMCFGR.EX field.

Type: `bool`

Default value: `true`

**`pmmir_ell_bus_slots`**

Largest value by which BUS\_ACCESS can increment over BUS\_CYCLES cycles. From v8.7 PMU extension.

Type: `uint16_t`

Default value: `0`

**`pmmir_ell_bus_width`**

Width, in bytes, of accesses counted by BUS\_ACCESS. From v8.7 PMU extension.

Type: `uint16_t`

Default value: `0x0`

**`pms_idr_max_size`**

Defines largest size for a single SPE record (rounded up to a power of 2).

Type: `uint8_t`

Default value: `6`

**`pmu-num_counters`**

Number of PMU counters implemented.

Type: `uint8_t`

Default value: `8`

**`pmu_async_exception_delay`**

Configure PMU asynchronous exception delay in CPU cycles (FEAT\_SEBEP).

Type: `uint32_t`

Default value: `0x0`

**`pmu_cycle_counter_counts_actual_cycles`**

If true and Timing annotation is enabled, PMU cycle counter counts actual cycles, otherwise counts instructions executed.

Type: `bool`

Default value: `false`

### **`pmu_has_chain_event`**

PMU (if present) implements event number 0x1e, CHAIN.

Type: `bool`

Default value: `true`

### **`pmu_precise_events`**

"Configure v9.4 Precise PMU events. {"pmu\_events":["SW\_INCR", "PC\_WRITE\_RETIRED", "BR\_RETIRED", "BR\_IND\_RETIRED", "BR\_RETURN\_RETIRED", "BR\_RETURN\_ANY\_RETIRED", "BR\_IND\_TAKEN\_RETIRED", "LD\_RETIRED", "ST\_RETIRED", "UNALIGNED\_LD\_ST", "INST\_RETIRED", "EXCEP\_TAKEN", "EXCEP\_RETURN", "CHAIN"]}".

Type: `string`

Default value: `N/A`

### **`pmu_threshold_bit_width`**

Implement FEAT\_PMUv3\_TH and if so the width of PMEVTYPER<n>\_ELO.TH in bits. 0, not implemented. 1-12 number of bits in PMEVTYPER<n>\_ELO.TH.

Type: `uint8_t`

Default value: `0`

### **`poe2_mmu_fault_far_aligned`**

If true, in case of an MMU fault on an access to IRT, DPOT or TTT, report FAR aligned to current translation granule or POE2 table size, whichever is smaller. By default, this is set to false - with FAR reporting the exact faulting VA.

Type: `bool`

Default value: `false`

### **`poison_range_end_addr`**

End PA of poisoned range.

Type: `uint64_t`

Default value: `0x0`

### **`poison_range_start_addr`**

Start PA of poisoned range.



Type: `uint64_t`

Default value: `0x0`

### **`preserve_cache_contents_over_warm_reset`**

Preserve cache contents over warm reset by ignoring the value of `ram_clear_on_reset_disable` signal in the cache.

Type: `bool`

Default value: `false`

### **`pseudo_fault_generation_feature_register`**

ARMv8.4 Standard Pseudo-fault generation feature register values. JSON schema for the parameter value is: [{"OF":false, "UC":false, "UEU":false, "UER":false, "UEO":false, "DE":false, "CE":0x0, "CI":false, "ER":false, "PN":false, "AV":false, "MV":false, "SYN":false, "R":false, "NA":false}, other\_pseudo-fault\_generating\_features\_register\_values]. Where OF, UC, UEU, UER, UEO, DE, CI, ER, PN, AV, MV, SYN, and R have valid false(NOT\_SUPPORTED) and true(FEATURE\_CONTROLLABLE), where CE can have 0(NOT\_SUPPORTED), 1(NONSPECIFIC\_CE\_SUPPORTED) and 3(TRANSIENT\_OR\_PERSISTENT\_CE\_SUPPORTED) and NA can have false(component fakes detection on next access) or true(component fakes detection spontaneously). Effective only when ERXFR's INJ field allows it or `has_ras_fault_injection` is true.

Type: `string`

Default value: `N/A`

### **`pstate_btype_on_illegal_eret`**

DEPRECATED. Please use `pstate_unknown_fields_on_illegal_eret` instead. This parameter will be ignored if both are used. In case of an illegal `eret`, what value to use to update `PSTATE.BTYPE`. 0: Set to 0, 1: Update with `SPSR_ELx.BTYPE`, 2: Unchanged.

Type: `uint8_t`

Default value: `0`

### **`pstate_pm_reset`**

Reset value of `PSTATE.PM`.

Type: `bool`

Default value: `false`

### **`pstate_ssbs_reset`**

Reset value of `pstate.ssbs`.

Type: `bool`

Default value: `false`

### **pstate\_ssbs\_type**

Implement speculative store bypass safe feature from ARMv8.5. 0, Not supported. 1, Supported without MSR/MRS access to SSBS (FEAT\_SSBS). 2, fully supported (FEAT\_SSBS2).

Type: `uint8_t`

Default value: 0

### **pstate\_unknown\_fields\_on\_illegal\_eret**

For an illegal eret, each unknown field in PSTATE can be set to "set\_to\_0", "set\_to\_spsr\_elx", or "unchanged". If a field is not specified but its relevant feature is enabled, it will default to "set\_to\_spsr\_elx". Fields for disabled features are ignored. Example: {"BTYP": "set\_to\_0", "DIT": "set\_to\_spsr\_elx", "PACM": "set\_to\_0", "SSBS": "set\_to\_spsr\_elx", "TCO": "unchanged", "UAO": "set\_to\_spsr\_elx"}.

Type: `string`

Default value: `N/A`

### **ptw\_latency**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

### **randomize\_unknowns\_at\_reset**

Will fill in unknown bits in registers at reset with random value using register\_reset\_data as seed, it overrides scramble\_unknowns\_at\_reset.

Type: `bool`

Default value: `false`

### **ras\_aderr\_anerr\_controls\_are\_same**

If true and FEAT\_ADERR and FEAT\_ANERR is implemented then ADERR and ANERR controls should always be set to the same value (FEAT\_ADERR) (FEAT\_ANERR).

Type: `bool`

Default value: `false`

### **ras\_err\_registers\_undef\_if\_no\_error\_records**

If true, all RAS error record registers, along with ERRSELR\_EL1, will be undefined if ERRIDR\_EL1 indicates that zero error records are implemented.

Type: `bool`

Default value: `false`

### **`ras_errselr_undef_if_no_error_records`**

If true, ERRSELR\_EL1 will be undefined if ERRIDR\_EL1 indicates that zero error records are implemented.

Type: `bool`

Default value: `false`

### **`ras_extra_configurations`**

Miscellaneous configurations for error records. An array of JSON objects. Note for ERXCTLR\_EL1 register it only allows to define the mask value for the IMPDEF fields, ie bits [63:32] and bit 1, but its reset value applies on all fields. Note for ERXMISCN masks - these are 64 bit masks covering the 64 bit registers ERXMISCN\_EL1. E.g. [{"Index": 0, "ERXMISCO\_mask": 0x0, "ERXMISCO\_reset": 0x0, "ERXMISC1\_mask": 0x0, "ERXMISC1\_reset": 0x0, "ERXMISC2\_mask": 0x0, "ERXMISC2\_reset": 0x0, "ERXMISC3\_mask": 0x0, "ERXMISC3\_reset": 0x0, "ERXCTLR\_EL1\_mask": 0x0, "ERXCTLR\_EL1\_reset": 0x0}, {"Index": 1, "ERXMISCO\_mask": 0x0, "ERXMISCO\_reset": 0x0, "ERXSTATUS\_IERR\_mask": 0x300}].

Type: `string`

Default value: `N/A`

### **`ras_frac`**

0, No additional feature implemented. 1, Additional ERXMISC, ERXPFG registers and FaultInjection trap from RAS v1.1. implemented.

Type: `uint8_t`

Default value: `0`

### **`ras_log2_fault_granule_size`**

Log2 of the RAS fault granule size in Bytes.

Type: `uint8_t`

Default value: `12`

### **`ras_mmap_address`**

Base address of memory mapped RAS Registers for each core on system bus. 0 means the RAS is not mapped, otherwise the address must be at least 4KB aligned or more depending upon the features implemented. JSON schema for the parameter value is: {"format": "all\_addrs\_are\_absolute\_wrt\_systembus", "cores": [{"ras": 0x0}, {"ras": 0x0}, {"ras": 0x0}, {"ras": 0x0}]}.

Type: `string`

Default value: `N/A`

### **`ras_pfg_clock_mhz`**

RAS Pseudo-Fault generation clock rate in MHz.

Type: `uint8_t`

Default value: `24`

### **`ras_pfg_err_opt`**

When accessing RAS Pseudo-fault Generation register, one of the following errors occurs under a certain condition: 0 - **NOP** Error, 1 - **UNDEF** Error, 2 - **OK**.

Type: `uint8_t`

Default value: `1`

### **`ras_report_aligned_pa_in_pfar`**

If true, the `PFAR_ELx` register reports the PA aligned to the RAS fault granule size on a sync external abort or SError exception.

Type: `bool`

Default value: `false`

### **`register_reset_data`**

Data used to fill register bits when they become **UNKNOWN** at reset.

Type: `uint64_t`

Default value: `0x0`

### **`register_reset_data_hi`**

Data used to fill the upper-half of 128-bit registers when the bits become **UNKNOWN** at reset.

Type: `uint64_t`

Default value: `0x0`

### **`report_iside_cmo_ifsr`**

fault info for an iside cache maintenance operation is reported in the IFSR.

Type: `bool`

Default value: `true`

**report\_second\_access\_align\_fault\_non\_atomic\_pair\_access**

If true, the non-atomic load/store pair accesses report the 2nd register as faulting address for an alignment fault. This is IMP-DEF behavior as defined in FEAT\_LRCPC3.

Type: `bool`

Default value: `false`

**report\_second\_access\_mmu\_fault\_non\_atomic\_pair\_access**

If true, the non-atomic load/store pair accesses report the 2nd register as faulting address for an MMU fault. This is IMP-DEF behavior as defined in FEAT\_LRCPC3.

Type: `bool`

Default value: `false`

**reported\_fp\_revision**

Purely cosmetic patch level value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored. Updates the FPSID.revision value.

Type: `int8_t`

Default value: -1

**reported\_patch\_level**

Purely cosmetic patch level value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: `int8_t`

Default value: -1

**reported\_revision\_number**

Purely cosmetic revision number value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: `int8_t`

Default value: -1

**reserved\_HMC\_SSC\_PAC\_treated\_disabled**

When DBG[B|W]CR.{HMC,SSC,PAC} bits configuration is reserved, this parameter controls whether breakpoints/watchpoints are treated as Disabled or not.

Type: `bool`

Default value: `false`

**restore\_fpsr\_on\_trapped\_fp\_exception**

If true, FPSR is restored to the value of the FPSR immediately before the instruction that generated the trapped floating-point exception.

Type: `bool`

Default value: `false`

**restriction\_on\_speculative\_execution**

Implements the ARMv8.5 security feature (Restrictions on the effects of speculation), ID\_AA64PFR0\_EL1.CSV2: 0: No disclosure whether branch targets trained in one context can affect speculative execution in a different context, 1: Branch targets trained in one context cannot affect speculative execution in a different hardware described context (SCXTNUM\_ELx not supported), 2: Branch targets trained in one context cannot affect speculative execution in a different hardware described context (SCXTNUM\_ELx supported) (FEAT\_CSV2, FEAT\_CSV2\_2), 3: FEAT\_CSV2\_3 is supported.

Type: `uint8_t`

Default value: 0

**restriction\_on\_speculative\_execution\_aarch32**

Implements the ARMv8.5 security feature (Restrictions on the effects of speculation), ID\_PFR0.CSV2: 0: No disclosure whether branch targets trained in one context can affect speculative execution in a different context, 1: Branch targets trained in one context cannot affect speculative execution in a different hardware described context, 2: Branch targets trained in one context cannot affect speculative execution in a different hardware described context or at a different address in the same hardware described context (FEAT\_CSV2, FEAT\_CSV2\_2).

Type: `uint8_t`

Default value: 0

**revision\_number**

Revision number of TRM implemented by the model. Changing the value can change the model behaviour. Visible as the Variant field in MIDR/MIDR\_EL1. Corresponds to the revision number X in rXpY.

Type: `uint8_t`

Default value: 0

**rgsr\_res0\_stateful**

Mask of RGSR **RES0** bits read of which return last written value.

Type: `uint64_t`

Default value: 0x0

**rme\_ctrl\_bits\_stateful\_level**

If `rme_support_level == 1`, specify the RME level that has control bits stateful in system registers. 0 - all until v9.2, 1 - v9.4 including FEAT\_RME\_GDI bits in MFAR\_EL3 and PFAR\_EL2.

Type: `uint8_t`

Default value: 0

**rme\_default\_mecid\_nonsecure**

Default MECID value for NON-SECURE PAS.

Type: `uint32_t`

Default value: 0x0

**rme\_default\_mecid\_realm**

Default MECID value for REALM PAS.

Type: `uint32_t`

Default value: 0x0

**rme\_default\_mecid\_root**

Default MECID value for ROOT PAS.

Type: `uint32_t`

Default value: 0x0

**rme\_default\_mecid\_secure**

Default MECID value for SECURE PAS.

Type: `uint32_t`

Default value: 0x0

**rme\_full\_is\_tagged\_nsh\_treated\_as\_tagged**

Whether a tagged NonShared memory attribute is treated as tagged or not. Does nothing if effective RME support is not full.

Type: `bool`

Default value: `false`

**rme\_level0\_gpt\_size**

The range of address space protected by each entry in the level 0 GPT (0->1GB 1->16GB, 2->64GB, 3->512GB).

Type: `uint8_t`

Default value: 0

### **`rme_mecid_width`**

Width of MECID in bits.

Type: `uint8_t`

Default value: 1

### **`rme_nsh_cacheable_is_shareable`**

If true, NSH cacheable becomes shareable cacheable (FEAT\_RME).

Type: `bool`

Default value: `false`

### **`rme_support_level`**

0 -> Realm management extension not implemented, 1 -> LEGACY\_TZ\_EN mode i.e. RME register fields are stateful but only supports secure/non-secure states, 2 -> Realm management extension fully implemented (FEAT\_RME).

Type: `uint8_t`

Default value: 0

### **`rnr_always_implemented`**

Always implement RMR\_ELx, RMR, or HRMR at the highest implemented exception level, even if that exception level cannot use both AArch32 and AArch64.

Type: `bool`

Default value: `false`

### **`rndr_rndrrs_seed`**

Initial seed for random engine used in RNDR register.

Type: `uint64_t`

Default value: 0x0

### **`s1_align_memtype_fault_prio_more_than_s2_perm_fault_on_s1_walk`**

If true, s1 alignment fault has priority over s2 permission faults.

Type: `bool`

Default value: `true`



**s1\_perm\_fault\_prio\_more\_than\_s2\_perm\_fault\_on\_s1\_walk**

If true, s1 permission fault has priority over s2 on s1 translation table walk permission faults.

Type: `bool`

Default value: `false`

**s1\_unsupported\_atomic\_fault\_for\_ls64\_prio\_more\_than\_s2\_perm\_fault**

If true, unsupported atomic/exclusive faults due to LS64 instructions at Stage 1 have higher priority than permission fault at Stage 2.

Type: `bool`

Default value: `false`

**scheduler\_mode**

Control the interleaving of instructions in this processor. 0, default long quantum. 1, low latency mode, short quantum and signal checking. 2, lock-breaking mode, long quantum with additional context switches near load-exclusive instructions. WARNING: This parameter is intended for validation purposes and may result in unwanted behaviour if altered!.

Type: `uint8_t`

Default value: 0

**scr\_nET\_writeable**

Whether SCR.nET is writeable. Writing to it is purely cosmetic (nET behavior not implemented).

Type: `bool`

Default value: `false`

**scramble\_unknowns\_at\_reset**

Will fill in unknown bits in registers at reset with `register_reset_data`.

Type: `bool`

Default value: `true`

**seerror\_clear\_delay**

Delay for clearing of SError if SError is level-triggered, in cpu cycles.

Type: `uint32_t`

Default value: `0x0`

**set\_mops\_option**

Set option for Armv8.8 SET(FEAT\_MOPS). 0, use default(i.e. use value configured through has\_mops\_option). 1, implemented using Option A. 2, implemented using Option B.

Type: `uint8_t`

Default value: 0

**set\_rasv10\_for\_armv84\_and\_higher**

ARMv8.4 mandates RAS System Architecture v1.1, but when there are no error records and FEAT\_DoubleFault is not implemented then there is no functional difference between the RAS System Architecture v1.0 (that is, the RAS extension as in pre-ARMv8.4 implementations) and the RAS System Architecture v1.1 (also known as FEAT\_RASv1p1). This flag if true will set the RAS ID to declare RAS v1.0 rather than RAS v1.1 for ARMv8.4 and higher implementations. If this is set and the core does not conform to the restrictions then this parameter is ignored.

Type: `bool`

Default value: `false`

**setg\_mops\_option**

Set option for Armv8.8 SETG(FEAT\_MOPS). 0, use default(i.e. use value configured through has\_mops\_option). 1, implemented using Option A. 2, implemented using Option B.

Type: `uint8_t`

Default value: 0

**skip\_trace\_on\_write\_to\_osecrr\_el1\_when\_oslock\_is\_unlocked**

If OSLSR\_EL1.OSLK == 0, then OSECRR\_EL1 returns an unknown value on reads and ignores writes. When true, also skips the traces on writes to OSECRR\_EL1 when OSLSR\_EL1.OSLK == 0.

Type: `bool`

Default value: `false`

**spe\_counter\_size**

Size of counter packet payload in Statistical Profiling Extension- 1, Counter packet payloads are 12-bit saturating counters- 2, Counter packet payloads are 16-bit saturating counters.

Type: `uint8_t`

Default value: 1

**spmu\_support\_level**

Implement System PMU: 0: Not supported, 1: v8.9 System PMU Extension is implemented (FEAT\_SPMU), 2: v9.5 System PMU2 Extension is implemented (FEAT\_SPMU2) values of this

parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **`spsr_el3_is_mapped_to_spsr_mon`**

Whether SPSR\_EL3 is mapped to AArch32 register SPSR\_mon.

Type: `bool`

Default value: `true`

### **`spsr_m4_res0`**

Whether SPSR\_ELx.M[4] bit should be **RES0** for AARCH64 only implementations.

Type: `bool`

Default value: `false`

### **`stage12_tlb_size`**

Number of stage1+2 tlb entries. Valid values are 0 or  $\geq 4$ .

Type: `uint32_t`

Default value: 0x4000

### **`stage1_tlb_size`**

Number of stage1 only tlb entries. Valid values are 0 or  $\geq 4$ .

Type: `uint32_t`

Default value: 0x0

### **`stage1_walkcache_size`**

Number of stage1 only walk cache entries.

Type: `uint32_t`

Default value: 0x0

### **`stage2_tlb_size`**

Number of stage2 only tlb entries.

Type: `uint32_t`

Default value: 0x0

**stage2\_walkcache\_size**

Number of stage2 only walk cache entries.

Type: `uint32_t`

Default value: `0x0`

**statistical\_profiling\_buffer\_alignment**

Statistical profiling alignment constraint for sample buffer.

Type: `uint16_t`

Default value: `0x1`

**statistical\_profiling\_datasrc\_payload\_size**

Statistical profiling size in bytes of Data Source packet' payloads.

Type: `uint8_t`

Default value: `1`

**statistical\_profiling\_random\_interval\_is\_separate**

Statistical profiling random interval gets added to the main timer interval(false) or (true) runs as separate timer.

Type: `bool`

Default value: `false`

**statistical\_profiling\_recommended\_min\_sampling**

Statistical profiling recommended minimum sampling interval.

Type: `uint16_t`

Default value: `0x100`

**stex\_fail\_suppress\_sync\_data\_aborts**

If true, synchronous data aborts are not reported if store exclusive fails.

Type: `bool`

Default value: `false`

**store\_excl\_fail\_tag\_check\_action**

Behavior of tag check by core when a store exclusive fails. 0, Tag check ignored, 1, Tag check done if exclusive fails by global monitor.

Type: `uint8_t`

Default value: 0

### **`strex_fail_can_hit_watchpoint`**

If true, a strex fail can hit watchpoint.

Type: `bool`

Default value: `false`

### **`stzgm_reports_fault_address_from_reg_arg`**

Which faulting address should be reported in FAR\_ELx on a failed STZGM: 0: the lowest aligned addr to DCZID-log2-block-size, 1: the addr held in the register argument, 2: if it is a tag-check fault, the addr aligned to DCZID-log2-block-size, otherwise the addr held in the register argument.

Type: `uint8_t`

Default value: 0

### **`supports_multi_threading`**

Sets the MPIDR.MT bit. Setting this to true hints the the cluster is multi-threading compatible.

Type: `bool`

Default value: `false`

### **`sve.clear_constrained_lanes`**

When a constrained vector length increases, previously inaccessible bits are set to zero. Possible values are: 0=never, 1=always, 2=if the register was written to while the vector length was constrained.

Type: `uint8_t`

Default value: 0

### **`sve.combine_movprfx_and_destructive`**

Attempt to combine the execution of MOVPRFX and the destructively-encoded instruction that follows it.

Type: `bool`

Default value: `false`

### **`sve.disable_speculative_accesses`**

All speculative memory accesses behave as though faulting, without accessing memory.

Type: `bool`

Default value: `false`

### **`sve.enable_at_reset`**

Start with system registers set up for Scalable Vector Extension use.

Type: `bool`

Default value: `false`

### **`sve.ffr_16b_pattern_UNKNOWN`**

A specific 16-bit **UNKNOWN** value that is used by parameter `force_UNKNOWN_to_ffr`.

Type: `uint16_t`

Default value: `0x0`

### **`sve.force_UNKNOWN_to_ffr`**

Governs behavior if WRFFR writes a non-monotonic value to FFR. Possible values are: 0 - Write non-canonical value to FFR, 1 - Overwrite FFR with a specific pattern of 16-bit **UNKNOWN** value. See `ffr_16b_pattern_UNKNOWN`, 2 - Clear all bits above first zero 3 - Set all bits after first one.

Type: `uint8_t`

Default value: `0`

### **`sve.fp_exception_report_lowest`**

If true, for multiple trapped FP exceptions, report the lowest lane in VECITR. Otherwise, report the highest.

Type: `bool`

Default value: `false`

### **`sve.fp_exception_set_tfv`**

Set `ESR_ELx.TFV` during FP exception. Trapped exception flags are valid.

Type: `bool`

Default value: `true`

### **`sve.fp_exception_set_vecitr`**

If true, set `ESR_ELx.VECITR` during FP exception. Otherwise, set **RES0**.

Type: `bool`

Default value: `false`

**sve.has\_b16b16**

Whether FEAT\_SVE\_B16B16 is implemented. Possible values are: 0 - Not implemented, 1 - Implemented if either FEAT\_SVE2 or FEAT\_SME2 is implemented.

Type: `uint8_t`

Default value: 0

**sve.has\_b16mm**

Whether FEAT\_SVE\_B16MM is implemented. Possible values are: 0 - Not implemented, 1 - Implemented if either FEAT\_SVE2 or FEAT\_SME2 is implemented.

Type: `uint8_t`

Default value: 0

**sve.has\_bfscale**

Whether FEAT\_SVE\_BFSCALE is implemented. Possible values are: 0 - Not implemented, 1 - Implemented if either FEAT\_SVE2 or FEAT\_SME2 is implemented.

Type: `uint8_t`

Default value: 0

**sve.has\_sme**

Whether SME is implemented (FEAT\_SME).

Type: `bool`

Default value: `false`

**sve.has\_sme2**

Whether SME2 is implemented (FEAT\_SME2).

Type: `bool`

Default value: `false`

**sve.has\_sme\_b16b16**

Whether FEAT\_SME\_B16B16 is implemented. Possible values are: 0 - Not implemented, 1 - Implemented if FEAT\_SME2 is implemented.

Type: `uint8_t`

Default value: 0

**sve.has\_sme\_f16f16**

Whether FEAT\_SME\_F16F16 is implemented. Possible values are: 0 - Not implemented, 1 - Implemented if FEAT\_SME2 is implemented.

Type: `uint8_t`

Default value: 0

**sve.has\_sme\_f64f64**

If SME is implemented, whether double-precision FMOPA and FMOPS are implemented.

Type: `uint8_t`

Default value: 1

**sve.has\_sme\_f8f16**

If SME2 is implemented, whether FEAT\_SME\_F8F16 is implemented.

Type: `uint8_t`

Default value: 1

**sve.has\_sme\_f8f32**

If SME2 is implemented, whether FEAT\_SME\_F8F32 is implemented.

Type: `uint8_t`

Default value: 1

**sve.has\_sme\_fa64**

Whether FEAT\_SME\_FA64 is implemented.

Type: `bool`

Default value: `false`

**sve.has\_sme\_i16i64**

If SME is implemented, whether instructions that accumulate 16-bit integer outer products into 64-bit integer tiles are implemented.

Type: `uint8_t`

Default value: 1

**sve.has\_sme\_lutv2**

Whether FEAT\_SME\_LUTv2 is implemented.



Type: `bool`

Default value: `false`

### **`sve.has_sme_mop4`**

Whether FEAT\_SME\_MOP4 is implemented. Possible values are 0 - Implemented if FEAT\_SME2p2 is implemented, 1 - Implemented if FEAT\_SME2p1 is implemented.

Type: `uint8_t`

Default value: 0

### **`sve.has_sme_priority_control`**

Whether SME Priority Control is implemented.

Type: `bool`

Default value: `true`

### **`sve.has_sme_tmop`**

Whether FEAT\_SME\_TMOP is implemented. Possible values are 0 - Implemented if FEAT\_SME2p2 is implemented, 1 - Implemented if FEAT\_SME2p1 is implemented.

Type: `uint8_t`

Default value: 0

### **`sve.has_ssve_aes`**

Indicates support for SVE2 and SME2 AES instructions when the PE is in Streaming SVE mode (FEAT\_SSVE\_AES).

Type: `uint8_t`

Default value: 0

### **`sve.has_ssve_bit_perm`**

Whether FEAT\_SSVE\_BitPerm is implemented. Possible values are 0 - Not Implemented, 1 - Implemented if FEAT\_SME2p1 is implemented.

Type: `uint8_t`

Default value: 0

### **`sve.has_ssve_fexpa`**

Whether FEAT\_SSVE\_FEXPA is implemented. Possible values are 0 - Implemented if FEAT\_SME2p2 is implemented, 1 - Implemented if FEAT\_SME2p1 is implemented.

Type: `uint8_t`

Default value: 0

### **sve.has\_sve2**

Whether SVE2 is implemented (FEAT\_SVE2).

Type: `bool`

Default value: `false`

### **sve.has\_sve2\_aes**

If SVE2 is implemented, whether AES instructions are implemented. Possible values are: 0 - not implemented, 1 - SVE2 AESE, AESD, AESMC, and AESIMC are implemented (FEAT\_SVE\_AES), 2 - Same as 1 but in addition SVE2 PMULLB and PMULLT with 64-bit source are implemented, 3 - Same as 2 but SVE2 64-bit source element PMLALB and PMLALT instruction variants are implemented (FEAT\_SVE\_PMULL128).

Type: `uint8_t`

Default value: 2

### **sve.has\_sve2\_bit\_perm**

If SVE2 is implemented, whether BitPerm instructions are implemented (FEAT\_SVE\_BitPerm).

Type: `bool`

Default value: `true`

### **sve.has\_sve2\_sha3**

If SVE2 is implemented, whether SHA3 instructions are implemented (FEAT\_SVE\_SHA3).

Type: `bool`

Default value: `true`

### **sve.has\_sve2\_sm4**

If SVE2 is implemented, whether SM4 instructions are implemented (FEAT\_SVE\_SM4).

Type: `bool`

Default value: `true`

### **sve.has\_sve\_bf16**

Whether SVE BFloat16 instructions are implemented.

Type: `bool`

Default value: `true`

**sve.has\_sve\_extended\_bf16**

Deprecated: to enable FEAT\_EBF16, use CPU parameter has\_ebf16. Whether Extended BFloat16 instructions are implemented. Possible values are: 0 - Disabled, 1 - Enabled if SME or SVE is implemented, 2 - Enabled if SME is implemented.

Type: `uint8_t`

Default value: 2

**sve.has\_sve\_f16f32mm**

Whether the SVE half-precision to single-precision Matrix Multiply instructions are implemented (FEAT\_F16F32MM).

Type: `bool`

Default value: `false`

**sve.has\_sve\_mm\_f32**

Whether the SVE FP32 Matrix Multiply instructions are implemented (FEAT\_F32MM).

Type: `bool`

Default value: `true`

**sve.has\_sve\_mm\_f64**

Whether the SVE FP64 Matrix Multiply instructions are implemented (FEAT\_F64MM).

Type: `bool`

Default value: `true`

**sve.has\_sve\_mm\_i8**

Whether the SVE Int8 Matrix Multiply instructions are implemented (FEAT\_I8MM).

Type: `bool`

Default value: `true`

**sve.movprfx\_unpredictable\_behavior**

Defines the behavior of MOVPRFX and the instruction it immediately precedes when the behavior is **CONSTRAINED UNPREDICTABLE**. Possible values are: 0 - UNDEF execution from MOVPRFX, 1 - MOVPRFX and second half of instruction executes as **NOP**, 2 - **NOP** MOVPRFX only, 3 - UNDEF execution from MOVPRFX unless otherwise trapped.

Type: `uint8_t`

Default value: 0

**sve.predicated\_sp\_align\_check\_behaviour**

Governs behavior of SP alignment checking for predicated memory accesses. Possible values are: 0 - Always perform, 1 - Skip if governing predicate is 0, 2 - Skip for contiguous accesses if governing predicate is 0, 3 - Skip for gather/scatter accesses if governing predicate is 0.

Type: `uint8_t`

Default value: 0

**sve.relax\_sme\_watchpoint\_matching\_16**

Whether memory accesses through Z and P registers in Streaming Mode and all accesses through ZA match watchpoints rounded to 16-byte alignment.

Type: `bool`

Default value: `false`

**sve.relax\_sve\_watchpoint\_matching\_16**

If FEAT\_DEBUGv8p9 is implemented, whether memory accesses through Z and P registers outside Streaming Mode match watchpoints rounded to 16-byte alignment.

Type: `bool`

Default value: `false`

**sve.sm\_tag\_checked**

Whether SME, SVE, and SIMD&FP load and store instructions executed when the PE is in Streaming SVE mode perform a Tag Check.

Type: `bool`

Default value: `true`

**sve.sme2\_version**

The version of SME2 if implemented. Possible values are: 0 - FEAT\_SME2, 1 - FEAT\_SME2p1, 2 - FEAT\_SME2p2, 3 - FEAT\_SME2p3.

Type: `uint8_t`

Default value: 0

**sve.sme\_highest\_implemented\_priority**

When SME Priority Control and SME2p2 are implemented, controls the highest implemented priority.

Type: `uint8_t`

Default value: 0

**sve.sme\_only**

If SME is implemented, whether SVE functionality is available only when SM=1.

Type: `bool`

Default value: `false`

**sve.sme\_ssve\_fp8\_support\_level**

If FEAT\_SME2 and FEAT\_FP8 are implemented, whether FP8 operations are supported in Streaming Mode where not implemented outside Streaming Mode. Possible values are: 0 - No support above FEAT\_FP8, 1 - FEAT\_SSVE\_FP8FMA, 2 - FEAT\_SSVE\_FP8DOT4, 3 - FEAT\_SSVE\_FP8DOT2.

Type: `uint8_t`

Default value: 0

**sve.sme\_vecleens\_implemented**

Which SME vector lengths are implemented. Represented as a bitfield where `bit[n]==1` implies SME vector length of  $128 \cdot 2^n$  bits is implemented.

Type: `uint8_t`

Default value: 7

**sve.smidr\_el1\_implementer\_val**

The value of SMIDR\_EL1.Implementer.

Type: `uint8_t`

Default value: 65

**sve.smidr\_el1\_nsmc\_val**

The value of SMIDR\_EL1.NSMC.

Type: `uint8_t`

Default value: 0

**sve.smidr\_el1\_revision\_val**

The value of SMIDR\_EL1.Revision.

Type: `uint8_t`

Default value: 0

**sve.smidr\_el1\_sh\_val**

The value of SMIDR\_EL1.SH.

Type: `uint8_t`

Default value: 0

**sve.sve2\_version**

The version of SVE2 if implemented. Possible values are: 0 - FEAT\_SVE2, 1 - FEAT\_SVE2p1, 2 - FEAT\_SVE2p2, 3 - FEAT\_SVE2p3.

Type: `uint8_t`

Default value: 0

**sve.sve\_dabt\_far\_behaviour**

Whether the FAR reported on a Data Abort is imprecise. Possible values are: 0 - FAR Precise, 1 - FAR not Precise on abort due to an SVE contiguous vector load/store or an SME load/store, 2 - As per 1, but only for predicated SVE/SME instructions, 3 - As per 1, but only for predicated SME/SVE load/store instructions that are executed in Streaming Mode.

Type: `uint8_t`

Default value: 0

**sve.sve\_wp\_far\_behaviour**

FAR reporting behavior on a Watchpoint debug exception. Possible values are: 0 - FAR Precise, 1 - FAR not Precise on abort due to an SVE contiguous vector load/store in Streaming SVE mode, or an SME load/store, 2 - FAR not valid on abort due to an SVE contiguous vector load/store in Streaming SVE mode, or an SME load/store, 3 - As per 1, but only for predicated SVE/SME instructions, 4 - As per 1, but only for predicated SME/SVE load/store instructions that are executed in Streaming Mode.

Type: `uint8_t`

Default value: 0

**sve.trace\_za\_tilewise**

Whether tile-wise accesses to ZA are traced tile-wise rather than array-wise. Note: if false, column-wise accesses cause an event for every vector in the tile.

Type: `bool`

Default value: `true`

**sve.undef\_invalid\_combined\_movprfx**

If a combined MOVPRFX is invalid, raise an UNDEF exception. Otherwise, **NOP** the second half. This parameter is deprecated.

Type: `bool`

Default value: `true`

**sve.unknown\_value**

Simulated value for a state that has an **UNKNOWN** value after reset.

Type: `uint64_t`

Default value: `0xdeaddeaddeaddead`

**sve.vecLEN**

SVE vector length in units of 64 bits.

Type: `uint8_t`

Default value: `8`

**sve.z\_reg\_on\_load\_fault\_behaviour**

Governs the behavior of destination Z-registers in case of a load fault. Possible values are: 0 - Register becomes **UNKNOWN**, 1 - Register is preserved.

Type: `uint8_t`

Default value: `0`

**sve.za\_on\_svl\_increase\_behaviour**

Controls the state of the previously inaccessible portion of the ZA registers on SVL increase. Possible values are: 0 - Retain values, 1 - Zero ZA.

Type: `uint8_t`

Default value: `0`

**sve.za\_tag\_checked**

Whether memory accesses due to SME LDR and STR instructions that access the SME ZA array perform a Tag Check.

Type: `bool`

Default value: `true`

**swp\_with\_xzr\_is\_st\_atomic**

If true, swp with dest as xzr is treated as store atomic.

Type: `bool`

Default value: `true`

**sync\_ext\_abort\_is\_sync\_serror**

Treat synchronous external aborts as synchronous SErrors (RASv8.9). 0, synchronous external abort. 1, synchronous serror.

Type: `bool`

Default value: `false`

**system\_pmu\_id**

When FEAT\_SPMU is implemented, indicates the largest value `s` to select a System PMU <`s`>.

Type: `uint8_t`

Default value: 0

**take\_ccfail\_tsc\_trap**

When `take_ccfail_undef=1` this parameter controls whether or not an SMC instruction that is trapped by HCR\_EL2.TSC but fails its condition code check generates a trap to EL2.

Type: `bool`

Default value: `false`

**take\_ccfail\_undef**

UNDEF exception is taken even if condition code check fails.

Type: `bool`

Default value: `true`

**tcr\_ps\_reserved\_value\_size**

Physical size treated when TCR.(I)PS is programmed with a reserved value. 0, 48 bits. 1, 52 bits. The parameter value is treated 0 if LPA is not supported.

Type: `uint8_t`

Default value: 0

**tcr\_tgx\_bit1\_stateful**

TCR.TGx[1] is stateful even without 16k granule support.



Type: `bool`

Default value: `false`

### **`tcr_txsz_undersize_should_fault`**

If large VA is not supported, Whether undersized TxSZ value should generate translation fault.

Type: `bool`

Default value: `true`

### **`tdosa_traps_osdlr_if_no_os_double_lock`**

MDCR\_EL\*.TDOSA enables trap on OSDLR\_EL1 and DBGOSDLR when OS double-lock is not implemented.

Type: `bool`

Default value: `true`

### **`tidcp_traps_el0_undef_imp_def`**

TIDCP has priority over UNDEF for accesses to **IMPLEMENTATION DEFINED** functionality from ELO.

Type: `bool`

Default value: `true`

### **`tlb_latency`**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

### **`tlbi_or_ic_invalid_xt`**

Behavior of TLBI and IC instructions that don't take Xt as an argument when Xt != 0b11111. 0: TLB and IC not UNDEF, 1: TLBI UNDEF, IC not UNDEF, 2: TLBI not UNDEF, IC UNDEF, 3: TLBI and IC UNDEF.

Type: `uint8_t`

Default value: `0`

### **`tlbi_stall_enabled`**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: `bool`

Default value: `false`

**tlbid\_nis**

Number of bits supported for IS TLBI Domains (FEAT\_TLBID). 0 - 16 : log(number of supported ISH tlbid domains).

Type: `uint8_t`

Default value: 0

**tlbid\_nos**

Number of bits supported for OS TLBI Domains (FEAT\_TLBID). 0 - 16 : log(number of supported OSH tlbid domains).

Type: `uint8_t`

Default value: 0

**tlbid\_nvis**

Number of bits supported for Virtual IS TLBI Domains (FEAT\_TLBID). 0 - 5 : log(number of supported Virtual ISH tlbid domains).

Type: `uint8_t`

Default value: 0

**tlbid\_nvos**

Number of bits supported for Virtual OS TLBI Domains (FEAT\_TLBID). 0 - 5 : log(number of supported Virtual OSH tlbid domains).

Type: `uint8_t`

Default value: 0

**tps\_support\_level**

Support for thread private state extension: 0 - Not implemented. 1 - Implemented at ELO only (FEAT\_TPS). 2 - Implemented at ELO, EL1 and EL2. (FEAT\_TPS, FEAT\_TPSP).

Type: `uint8_t`

Default value: 0

**trace\_full\_simd\_reg\_with\_nep**

Whether full simd register is traced even if partial update is done when FPCR.NEP=1.

Type: `bool`

Default value: `false`

**trace\_has\_sysreg\_access**

ETM trace registers support access via system registers.

Type: `bool`

Default value: `true`

**trace\_icc\_registers\_as\_icv\_when\_redirected**

If true, update trace with ICV, instead of ICC when ICV registers are accessed depending on the core state.

Type: `bool`

Default value: `false`

**trace\_physical\_registers\_when\_host\_virtualisation\_enabled**

Trace sysreg accesses to physical register accessed (0=disabled, 1=Trace only ELR/SPSR\_EL1 as ELR/SPSR\_EL2, 2=Trace all redirected registers as physical registers affecting all features inc. the host virtualised registers.

Type: `uint8_t`

Default value: `0`

**trace\_xzr\_in\_core\_regs64\_trace**

Whether CORE\_REGS64\_READ traces XZR and WZR input registers.

Type: `bool`

Default value: `true`

**trap\_dc\_cmo\_to\_pou\_if\_nop**

Whether traps to DC CMO operations to PoU are ignored if the same is treated as **NOP**.

Type: `bool`

Default value: `true`

**trap\_ic\_cmo\_to\_pou\_if\_nop**

Whether traps to IC CMO operations to PoU are ignored if the same is treated as **NOP**.

Type: `bool`

Default value: `true`

**trap\_reserved\_group3\_id\_regs**

Whether setting HCR\_EL2.TID3 traps reserved group3 id registers.

Type: `bool`

Default value: `false`

### **trbe\_cmod**

TRBE Customer Modified.

Type: `uint8_t`

Default value: `0`

### **trbe\_des**

Designer, JEP106 identification code.

Type: `uint16_t`

Default value: `0x0`

### **trbe\_external\_abort\_handling**

Describes how the PE manages External aborts on writes made by the Trace Buffer Unit to the trace buffer. (0->External abort is reported to TRBE. From Armv9.3, the value 0 is not permitted and will be 1 if Armv9.3 is implemented. 1-> External abort is ignored. 2->The External abort generates an SError and the error is not reported to TRBE.).

Type: `uint8_t`

Default value: `0`

### **trbe\_has\_hardware\_translation\_table\_update**

If true, address translation performed by the Trace Buffer Extension manages the Access Flag and dirty state.

Type: `bool`

Default value: `true`

### **trbe\_implemented\_version**

Trace Buffer Extension implemented version, 1: FEAT\_TRBE implemented (Armv9.0), 2: FEAT\_TRBEv1p1 and FEAT\_TRBE\_EXC are implemented.

Type: `uint8_t`

Default value: `1`

### **trbe\_mpam**

TRBE MPAM support.

Type: `uint8_t`

Default value: 0

**trbe\_part**

Part number.

Type: `uint16_t`

Default value: 0x0

**trbe\_partid\_max**

Largest permitted TRBDEVID1.PARTID value.

Type: `uint16_t`

Default value: 0x0

**trbe\_pmg\_max**

Largest permitted TRBDEVID1.PMG value if FEAT\_MPAMv2 is implemented, otherwise 0xff.

Type: `uint16_t`

Default value: 0x0

**trbe\_revand**

TRBE component minor revision.

Type: `uint8_t`

Default value: 0

**trbe\_revision**

TRBE architecture revision.

Type: `uint8_t`

Default value: 0

**trbe\_stop\_on\_misaligned\_pointers**

If true, the Trace Buffer Extension will stop tracing if a buffer pointer is not aligned to TRBIDR\_EL1.Align.

Type: `bool`

Default value: `false`

**trbe\_trbptr\_el1\_has\_res0**

If true, when TRBIDR\_EL1.Align is not zero, bits [M-1:0] in TRBPTR\_EL1 are not stateful.

Type: `bool`

Default value: `false`

#### **`treat-dcache-cmos-to-occ-as-nop`**

Implement CMOs to Outer cache level as **NOP**.

Type: `bool`

Default value: `false`

#### **`treat-dcache-cmos-to-poc-as-nop`**

Whether dcache maintenance operations to the point of coherency are required for instruction to data coherence. 0 - Clean/Invalidate ops required, 1 - Clean/Invalidate ops not required and cannot generate faults, 2 - Clean/Invalidate ops not required but can generate faults.

Type: `uint8_t`

Default value: 0

#### **`treat-dcache-cmos-to-pou-as-nop`**

Whether dcache invalidation to the point of unification is required for instruction to data coherence. 0 - Invalidate ops required, 1 - Invalidate ops not required and cannot generate faults, 2 - Invalidate ops not required but can generate faults.

Type: `uint8_t`

Default value: 0

#### **`treat-dcache-invalidate-as-clean-invalidate`**

Treat data cache invalidate operations as clean and invalidate.

Type: `bool`

Default value: `false`

#### **`treat-icache-cmos-to-pou-as-nop`**

If `has_coherent_icache` is true, whether instruction cache invalidation operations to PoU which are treated as **NOP** can generate fault. 0 - cannot generate faults, 1 - can generate faults.

Type: `uint8_t`

Default value: 0

#### **`treat_PAC_as_NOP`**

Non-architecture parameter. Treat Pointer Authentication as **NOP**. When the parameter is true behaviour of FEAT\_PAAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions

are **NOP**. Branch, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAAuth traps.

Type: `bool`

Default value: `false`

#### **`treat_forced_normal_as_device_for_excl_atomics`**

Whether exclusive/atomic access is supported in same manner as access to device if stage1 is Device memory and final memory attribute forced to normal by FWB.

Type: `bool`

Default value: `false`

#### **`treat_pld_as_nop`**

If true, treat PLD as **NOP**.

Type: `bool`

Default value: `false`

#### **`treat_pli_as_nop`**

If true, treat PLI as **NOP**.

Type: `bool`

Default value: `false`

#### **`treat_wfi_wfe_as_nop`**

If true, never go into wait state for WFI or WFE instructions.

Type: `bool`

Default value: `false`

#### **`truncate_pc_on_illegal_exception_return_to_aarch32`**

On Illegal ERET to AArch32, truncate PC to 32-bits.

Type: `bool`

Default value: `true`

#### **`ttt_fetch_fault_report_type`**

Behaviour for TTT fetch faults: 0, report as data abort. 1, report as instruction abort.

Type: `bool`

Default value: `false`

### **tune\_spe\_cache\_events**

"Set the percentage of cache event bits set on a load instruction. Supplying any valid, non-empty JSON makes PMSNEVFR\_EL1[23:19] stateful. Each field if present, must be between 5-95. Omitted fields default to 0. Example: {'l2\_dcache\_access':20, 'l2\_dcache\_miss':20, 'dcache\_modified':20, 'recently\_fetched':20, 'data\_snooped':20}"

Type: `string`

Default value: `N/A`

### **undef\_ccsidr2\_access\_for\_unimplemented\_aarch32**

Whether access to CCSIDR2 is undef if AArch32 is implemented or not at EL1.

Type: `bool`

Default value: `false`

### **unification-level**

Level of Unification Inner Shareable for the cache hierarchy.

Type: `uint8_t`

Default value: `1`

### **unification-uniprocessor-level**

Level of Unification Uniprocessor for the cache hierarchy.

Type: `uint8_t`

Default value: `1`

### **unpred\_LSE128\_overlap**

Constrained unpredictable behaviours for 128-bit LSE overlap. 1 Constraint\_UNDEF, 2 Constraint\_NOP.

Type: `uint8_t`

Default value: `1`

### **unpred\_brb\_iall\_or\_inj\_invalid\_xt\_behave\_as\_undef**

If true, BRB IALL/INJ instruction will behave as **UNDEFINED** if Xt != 0b11111.

Type: `bool`

Default value: `false`



**unpred\_brbe\_next\_branch\_cycle\_count\_unknown**

If true, cycle count value for the next BRBE branch record after BRB INJ execution outside prohibited region is unknown.

Type: `bool`

Default value: `false`

**unpred\_clear\_ISV\_for\_exception\_before\_software\_step**

Whether ESR\_ELx.ISV bit is cleared/set, when it is constrained unpredictable due to a different exception before a software step exception.

Type: `bool`

Default value: `false`

**unpred\_edscr\_ns\_set\_unknown\_bit**

Unknown(x) bit in NS field in EDSCR can be configure to 0 or 1.

Type: `bool`

Default value: `false`

**unpred\_edscr\_rw\_unknown\_bits\_read\_as\_1**

Unknown(x) bits in RW field in EDSCR are read as 1 instead of 0.

Type: `bool`

Default value: `false`

**unpred\_edscr\_status\_read\_as\_no\_syndrome**

Controls the choice of EDSCR.STATUS bit-values, when it is constrained unpredictable behaviour due to a different exception before a halting step debug event.

Type: `bool`

Default value: `false`

**unpred\_extdbg\_unknown\_bits**

Data used to fill only in **UNKNOWN** bit-fields of external debug registers e.g., EDPFR and EDDFR.

Type: `uint64_t`

Default value: `0x0`

**unpred\_load\_single\_reg\_overlap\_with\_wb**

Constrained unpredictable behaviours for single load with writeback(might impact certain load pair instructions) 0 Constraint\_WBSUPPRESS, 1 Constraint\_UNDEF, 2 Constraint\_NOP.

Type: `uint8_t`

Default value: 0

**unpred\_mrsmsr\_currentlymapped\_undef**

**UNPREDICTABLE** register access (accessible from current mode using different instruction) modeled as **NOP** when false and **UNDEF** when true.

Type: `bool`

Default value: `false`

**unpred\_mrsmsr\_protfailed\_undef**

**UNPREDICTABLE** register access (not accessible from current PL and security state) modeled as **NOP** when false and **UNDEF** when true.

Type: `bool`

Default value: `false`

**unpred\_mte\_stzgm\_tag\_operation\_before\_data**

Whether Tag operations are performed before data operations for an STGZM instruction.

Type: `bool`

Default value: `true`

**unpred\_mte\_tag\_read\_when\_ata\_controls\_are\_zero\_or\_untagged\_attr**

Constrained unpredictable for MTE tag read when ATA controls are 0 or untagged attribute. false, Read as zero. true, Permitted to generate an external abort if a read of data from the same address would generate an external abort.

Type: `bool`

Default value: `false`

**unpred\_mte\_tag\_store\_data\_cache\_instr\_to\_device\_mem\_as\_alignment\_fault**

Constrained unpredictable choice for MTE instructions which store tags (on DC instructions) to memory locations marked as Device. false, Storing the data, if any, to the locations. true, Generating an Alignment Fault.

Type: `bool`

Default value: `false`

**unpred\_mte\_tag\_store\_to\_device\_mem\_as\_alignment\_fault**

Constrained unpredictable choice for STZGM instruction which store tags to memory locations marked as Device. false, Storing the data, if any, to the locations. true, Generating an Alignment Fault.

Type: `bool`

Default value: `false`

**unpred\_nested\_virtualization\_nv\_behaviour**

Constrained unpredictable choices for HCR\_EL2.NV=0 and HCR\_EL2.NV1=1 with respect to nested virtualization- 0, Behave as defined in the specification as per bit values- 1, Behave as if HCR\_EL2.NV=1 and HCR\_EL2.NV1=1 for all purpose other than reading back HCR\_EL2.NV- 2, Behave as if HCR\_EL2.NV=0 and HCR\_EL2.NV1=0 for all purpose other than reading back HCR\_EL2.NV1.

Type: `uint8_t`

Default value: 0

**unpred\_par\_attr\_returns\_mair**

If true, PAR\_EL1.ATTR represents the memory attributes as per the MAIR value instead of the ones in the descriptor.

Type: `bool`

Default value: `false`

**unpred\_poe2\_va\_ress\_mismatch**

Constrained unpredictable choices when bits marked as RESS do not all have the same value for POE2 registers containing a VA - 0, Generating a translation abort on use of the register- 1, Reserved sign extended bits of the register are same as bit[52] or bit[48] based on if large VA is supported or not, for all purposes other than reading back the register- 2, Reserved sign extended bits of the register are same as bit[52] or bit[48] based on if large VA is supported or not, for all purposes .

Type: `uint8_t`

Default value: 0

**unpred\_s2\_hw\_dirty\_update\_on\_atomic\_wo\_read\_perm\_fault**

Constrained unpredictable behavior for atomic instructions that generate a stage 2 permission fault only due to lack of read permission, on a stage 2 writable-clean descriptor. If true, hardware is allowed to update the stage 2 dirty state; else, the dirty update is suppressed.

Type: `bool`

Default value: `false`

**unpred\_sctlr\_c\_0\_taggable\_behaviour**

Controls unpredictable effects when SCTLTR\_ELx.C=0 for a stage 1 translation regime on whether memory is treated as taggable. Values: 0=Tagged, 1=Untagged but forced to Tagged when FWB=1 and stage 2 restores WB, 2 = Untagged.

Type: uint8\_t

Default value: 2

**unpred\_store\_exclusive\_base\_overlap**

Constrained unpredictable behaviours for store exclusive when s==n. 0 Constraint\_NONE, 1 Constraint\_UNDEF, 2 Constraint\_NOP.

Type: uint8\_t

Default value: 0

**unpred\_store\_pair\_and\_single\_reg\_overlap\_with\_wb**

Constrained unpredictable behaviours for pair and single store with writeback(doesn't cover store exclusive) 0 Constraint\_NONE, 1 Constraint\_UNDEF, 2 Constraint\_NOP.

Type: uint8\_t

Default value: 0

**unpred\_tchange\_tenter\_and\_texit\_behaviour**

TCHANGE, TENTER and TEXTIT unpredictable behaviour in debug state. 0, **NOP**. 1, Undefined. 2, Execute as in non-debug state.

Type: uint8\_t

Default value: 0

**unpred\_tlbi\_not\_in\_monitor\_mode**

Constrained unpredictable behaviors for AArch32 TLBI instructions executed in secure privileged mode other than Monitor mode. 0: Preferred behavior (default), 1: UNDEF, 2: **NOP**, 3: execute as if had been executed in Monitor mode.

Type: uint8\_t

Default value: 0

**unpred\_tps\_range**

When TPMINn\_ELx > TMAXn\_ELx: 0 - Prevent all accesses. 1 - Allow all accesses. 2 - Wrap around.

Type: uint8\_t

Default value: 0

#### **unpred\_tps\_va\_ress\_mismatch**

Constrained unpredictable choices when bits marked as RESS do not all have the same value for TPS registers containing a VA - 0, Generating a translation abort on use of the register- 1, Reserved sign extended bits of the register are same as bit[52] or bit[48] based on if large VA is supported or not, for all purposes other than reading back the register- 2, Reserved sign extended bits of the register are same as bit[52] or bit[48] based on if large VA is supported or not, for all purposes .

Type: `uint8_t`

Default value: 0

#### **unpred\_tsize\_aborts**

Behaviour when TSize is out of range. 0, force into range. 1, translation fault, forces `unpred_tsize_pamax_aborts` to 1.

Type: `bool`

Default value: `false`

#### **unpred\_tsize\_pamax\_aborts**

Behaviour when stage 2 TSize exceeds the physical address size (or 40bits, from AArch32). 0, force into range. 1, translation fault. Ignored if `unpred_tsize_aborts` is 1.

Type: `bool`

Default value: `false`

#### **unpred\_vnocr\_el2\_ress\_mismatch**

Constrained unpredictable choices when bits marked as RESS do not all have the same value for VNCR\_EL2 - 0, Generating an EL2 translation regime translation abort on use of the VNCR\_EL2 register- 1, Reserved sign extended bits of VNCR\_EL2 are same as bit[52] or bit[48] based on if large VA is supported or not, for all purposes other than reading back the register- 2, Reserved sign extended bits of VNCR\_EL2 are same as bit[52] or bit[48] based on if large VA is supported or not, for all purposes .

Type: `uint8_t`

Default value: 0

#### **unpred\_zero\_spsr\_btype**

Constrained unpredictable control to make SPSR\_ELx.BTYPE 0 instead of PSTATE.BTYPE on synchronous exceptions other than Software Step, PC alignment fault, Instruction Abort, Breakpoint or Address Matching Vector Catch, Illegal Execution State, BRK instruction, Branch Target.

Type: `bool`

Default value: `true`

### **unpredictable\_exclusive\_abort\_memtype**

Cause MMU abort if exclusive access is not supported in certain memory type (0=exclusives allowed in all memory types, 1=exclusives abort in Device memory types, 2=exclusives abort in any type other than WB inner cacheable).

Type: `uint8_t`

Default value: 0

### **unpredictable\_hvc\_behaviour**

HVC unpredictable behaviour. 0, UNDEF. 1, **NOP**.

Type: `uint8_t`

Default value: 0

### **unpredictable\_smc\_behaviour**

SMC unpredictable behaviour. 0, UNDEF. 1, **NOP**.

Type: `uint8_t`

Default value: 0

### **unpredictable\_wfet\_and\_wfit\_behaviour**

WFET and WFIT unpredictable behaviour in debug state. 0, **UNDEFINED**. 1, **NOP**.

Type: `uint8_t`

Default value: 1

### **unsupported\_atomic\_fault\_type**

Type of fault reported on unsupported atomic access. 0 = external abort if any reported by interconnect, 1 = precise unsupported atomic fault, 2 = precise external abort, 3 = imprecise external abort.

Type: `uint8_t`

Default value: 0

### **unsupported\_fp8\_format\_behaviour**

Behaviour when FPMR.{F8S1,F8S2,F8D} are programmed to a reserved value 0->FP8 Inputs are treated as a signalling NaN, FP8 outputs are 0xFF 1->Format is treated as FPMR.{F8S1,F8S2,F8D} & 0x1.

Type: `uint8_t`

Default value: 0

### **unsupported\_hw\_update\_fault\_type**

Type of abort reported when hw update to descriptor is done using unsupported memtype (0=No abort, 1=IMPDEF abort caused by memtype, 2=Sync external abort).

Type: `uint8_t`

Default value: 0

### **use\_architectural\_names**

Use names SP/LR/PC instead of R13/R14/R15.

Type: `bool`

Default value: `false`

### **use\_sif\_to\_compute\_pan**

Where FEAT\_PAN3 is implemented, whether SCR\_EL3.SIF bit is used to determine instruction access permission for the purpose of PAN.

Type: `bool`

Default value: `false`

### **use\_stage1\_sh\_as\_input\_to\_stage2**

IMPDEF case of whether to use stage1 shareability or OuterShareable as input to stage2 if stage1 is Device memtype.

Type: `bool`

Default value: `false`

### **use\_tlb\_contig\_hint**

Translation table entries with the contiguous hint bit set generate large TLB entries.

Type: `bool`

Default value: `false`

### **user\_defined\_rom\_table\_debug\_power\_config**

User defined ROM Table debug power domains for ED,CTI,PMU and TRACE, and DBGPCR configuration. The "version" field and "cores" array are mandatory. The "dbgpcr" array, if provided, must contain unique integers in the range [0, 31] describing which debug power domains have power control implemented. The "rom" and "dbgpcr" fields in objects in the "cores" array are only allowed when 'debug\_rom\_is\_flat' is false. All power domain ID fields ("rom", "ed/pmu", "cti", "etm") must be in the range [0, 31]. The "ed/pmu" field is mandatory. Example JSON for a hierarchical

debug ROM layout: '{"version": 0, "dbgpcr": [0, 1], "cores":[{"dbgpcr": [1, 31], "rom": 0, "ed/pmu": 0, "cti": 31, "etm": 1}, {"ed/pmu": 0}]}'.

Type: `string`

Default value: `N/A`

### **`vmte_support_level`**

Specify the Virtual Tagging Base Feature support level:- 0 Not Implemented- 1 Implement Virtual Tagging and Checking controls and instructions, but not enablement (FEAT\_VMTE+FEAT\_VMTETC)- 2 As per 1 but controls to enable Virtual tagging and tag checking are not ignored, and access to tags in memory is enabled (FEAT\_VMTEE+FEAT\_VMTETCE).

Type: `uint8_t`

Default value: `0`

### **`vpu_datapath_width`**

VPU data path width.

Type: `uint8_t`

Default value: `128`

### **`walk_cache_latency`**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

### **`warn_for_dbgwcr_reserved_values_with_razwi_bits`**

Display a warning when DBGWCR is programmed with a reserved value even if some bits(e.g. HMC) are **RAZ/WI**.

Type: `bool`

Default value: `true`

### **`warn_unpredictable_in_v7`**

If true, behaviour which is unpredictable in V7 yet is predictable in V8 will produce a warning.

Type: `bool`

Default value: `true`



**watchpoint-log2secondary\_restriction**

log2 size of secondary restriction of FAR/EDWAR possible values on watchpoint hit for load/store operations.

Type: `uint8_t`

Default value: 0

**wfe\_wakeup\_delay**

Configure WFE wakeup delay in CPU cycles.

Type: `uint32_t`

Default value: 0x0

**wfi\_wakeup\_delay**

Configure WFI wakeup delay in CPU cycles.

Type: `uint32_t`

Default value: 0x0

**wnr\_is\_read\_for\_s2f\_on\_s1\_atomic\_instr\_fault**

Whether WnR is 0 for stage2 fault on stage1 for atomic instructions.

Type: `bool`

Default value: `false`

**wnr\_is\_read\_for\_s2f\_on\_s1\_dbm\_update**

Whether WnR is 0 for stage2 fault on stage1 descriptor dbm update.

Type: `bool`

Default value: `false`

**wp\_ignores\_dbm\_update**

If true, dbm update is ignored on watchpoint hit.

Type: `bool`

Default value: `false`

**wp\_num\_reporting**

When reporting of the watchpoint number on Watchpoint Exceptions and Debug Events is performed - When FEAT\_Debugv8p9 is implemented or otherwise required - When FEAT\_Debugv8p9 or FEAT\_SME is implemented.

Type: `uint8_t`

Default value: 0

## 3.4 AEMvA\_DSUCT

Defined in `LISA/AEMvA_DSUCT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Alpha support

For an explanation of the quality levels, see [Quality level definitions](#).

### About AEMvA\_DSUCT

AEMvA\_DSUCT CPU component.

### Iris and MTI instances for AEMvA\_DSUCT

This model has the following Iris instances:

Name	Instance type
AEMvA_DSUCT	Cluster_ARM_AEMvA_DSU_MP
AEMvA_DSUCT.AMU	PVBusLogger
AEMvA_DSUCT.AMU.mapper	PVBusMapper
AEMvA_DSUCT.DAP	PVBusLogger
AEMvA_DSUCT.DAP.mapper	PVBusMapper
AEMvA_DSUCT.DSU	C1-DSU
AEMvA_DSUCT.DSU.PPU_cluster	PPUv1
AEMvA_DSUCT.DSU.PPU_cluster.busslave	PVBusSlave
AEMvA_DSUCT.DSU.PPU_core0	PPUv1
AEMvA_DSUCT.DSU.PPU_core0.busslave	PVBusSlave
AEMvA_DSUCT.DSU.mpam_busslave	PVBusSlave
AEMvA_DSUCT.DSU.shared_cache	PVCache
AEMvA_DSUCT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
AEMvA_DSUCT.DSU.shared_cache.upstream[X] (where X = 0-4)	PVBusSlave
AEMvA_DSUCT.DSU.utility_slave[0]	PVBusSlave
AEMvA_DSUCT.MMAP	PVBusLogger
AEMvA_DSUCT.MMAP.mapper	PVBusMapper
AEMvA_DSUCT.RAS	PVBusLogger
AEMvA_DSUCT.RAS.mapper	PVBusMapper
AEMvA_DSUCT.cpu0	ARM_AEM-A_DSU_MP
AEMvA_DSUCT.cpu0.UTLB	TLB

Name	Instance type
AEMvA_DSUCT.cpu0.debug_rom	debug_rom
AEMvA_DSUCT.cpu0.dtlb	TLB
AEMvA_DSUCT.cpu0.l1dcache	PVCache
AEMvA_DSUCT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
AEMvA_DSUCT.cpu0.l1dcache.upstream[0]	PVBusSlave
AEMvA_DSUCT.cpu0.l1icache	PVCache
AEMvA_DSUCT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
AEMvA_DSUCT.cpu0.l1icache.upstream[0]	PVBusSlave
AEMvA_DSUCT.cpu0.l2cache	PVCache
AEMvA_DSUCT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
AEMvA_DSUCT.cpu0.l2cache.upstream[Z] (where Z = 0-1)	PVBusSlave
AEMvA_DSUCT.ext_bus	PVBusLogger
AEMvA_DSUCT.ext_bus.mapper	PVBusMapper
AEMvA_DSUCT.gic_cpuid_decoder_cluster	GICv3CPUInterfaceDecoder
AEMvA_DSUCT.global_debug_rom	debug_rom
AEMvA_DSUCT.secondary_debug_rom	debug_rom

This model has the following MTI trace components:

Name	Component type
AEMvA_DSUCT.AMU	PVBusLogger
AEMvA_DSUCT.AMU.mapper	PVBusMapper
AEMvA_DSUCT.DAP	PVBusLogger
AEMvA_DSUCT.DAP.mapper	PVBusMapper
AEMvA_DSUCT.DSU	C1-DSU
AEMvA_DSUCT.DSU.PPU_cluster	PPUv1
AEMvA_DSUCT.DSU.PPU_cluster.busslave	PVBusSlave
AEMvA_DSUCT.DSU.PPU_core0	PPUv1
AEMvA_DSUCT.DSU.PPU_core0.busslave	PVBusSlave
AEMvA_DSUCT.DSU.mpam_busslave	PVBusSlave
AEMvA_DSUCT.DSU.shared_cache	PVCache
AEMvA_DSUCT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
AEMvA_DSUCT.DSU.shared_cache.upstream[X] (where X = 0-4)	PVBusSlave
AEMvA_DSUCT.DSU.utility_slave[0]	PVBusSlave
AEMvA_DSUCT.MMAP	PVBusLogger
AEMvA_DSUCT.MMAP.mapper	PVBusMapper
AEMvA_DSUCT.RAS	PVBusLogger
AEMvA_DSUCT.RAS.mapper	PVBusMapper
AEMvA_DSUCT.cpu0	ARM_AEM-A_DSU_MP
AEMvA_DSUCT.cpu0.UTLB	TLB
AEMvA_DSUCT.cpu0.l1dcache	PVCache

Name	Component type
AEMvA_DSUCT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
AEMvA_DSUCT.cpu0.l1dcache.upstream[0]	PVBusSlave
AEMvA_DSUCT.cpu0.l1licache	PVCache
AEMvA_DSUCT.cpu0.l1licache.downstream[0].pvbusmaster	PVBusMaster
AEMvA_DSUCT.cpu0.l1licache.upstream[0]	PVBusSlave
AEMvA_DSUCT.cpu0.l2cache	PVCache
AEMvA_DSUCT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
AEMvA_DSUCT.cpu0.l2cache.upstream[Z] (where Z = 0-1)	PVBusSlave
AEMvA_DSUCT.ext_bus	PVBusLogger
AEMvA_DSUCT.ext_bus.mapper	PVBusMapper
AEMvA_DSUCT.gic_cpuif_decoder_cluster	GiCv3CPUInterfaceDecoder

### Ports for AEMvA\_DSUCT

Port	Direction	Protocol	Description
acp_s	slave	PVBus	AXI ACP subordinate port.
AEND0MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND2MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND3MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART2MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	slave	Signal	CHI defined pins.
broadcastcachemaint	slave	Signal	ACE defined pins.
broadcastouter	slave	Signal	ACE defined pins.
broadcastpersist	slave	Signal	CHI defined pins.
cfgend	slave	Signal	This signal is for EE bit initialisation.
cfgsdisable	slave	Signal	This signal disables write access to some secure Interrupt Controller registers.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcsn_pchannel	master	PChannel	Cluster PCSM signal

Port	Direction	Protocol	Description
cluster_powerdown_out	master	Signal	DEPRECATED - An output from PPUs that informs thermal controller of the cluster power info
cluster_ppu_hw_stat	master	Value	Cluster PPU power state
clusterid	slave	Value	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set ManagerID64 of the core.
clusterpmuirq	master	Signal	DynamlQ pmu irq
cme_pcs_m_pchannel	master	PChannel	Cluster PCSM signal
cmeerirq	master	Signal	RAS cme err irq
cmefaultirq	master	Signal	RAS cme fault irq
CNTHPIRQ	master	Signal	Timer signals to SOC.
CNTHPSIRQ	master	Signal	Timer signals to SOC
CNTHVIRQ	master	Signal	Timer signals to SOC.
CNTHVSIRQ	master	Signal	Timer signals to SOC
CNTPNSIRQ	master	Signal	Timer signals to SOC.
CNTPSIRQ	master	Signal	Timer signals to SOC.
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.
CNTVIRQ	master	Signal	Timer signals to SOC.
commirq	master	Signal	Interrupt signal from debug communications channel.
core_clk_in	slave	ClockSignal	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
core_pcs_m_pchannel	master	PChannel	Core PCSM signals
core_powerdown_out	master	Signal	DEPRECATED - An output from PPUs that informs thermal controller of the core power info
core_ppu_hw_stat	master	Value	Notify the power state change inside the PPU.
coreerrirq	master	Signal	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq	master	Signal	Fault indicator for a detected 1 or 2 bit ECC error
coreinstrrun	master	Signal	Core Running State
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
ctidbgirq	master	Signal	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	slave	Signal	External debug interface.
dbgnopwrdown	master	Signal	No power-down request.
dbgpwrupreq	master	Signal	Debug power up request.

Port	Direction	Protocol	Description
defaultMP	slave	Signal	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main manager interface ports.
dev_debug_s	slave	PVBus	External debug interface.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.
gicreset	master	Signal	An output from PPU that can be used to reset GICCLK domain components
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	slave	PVBus	External debug interface.
pmbirq	master	Signal	Interrupt signal from the statistical profiling unit.
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	master	Signal	PPU cluster interrupt.
ppu_cluster_isolate	master	Signal	PPU Cluster Domain isolation control.
ppu_cluster_wakerequest	slave	Signal	PPU cluster wake request signal.
ppu_cme_irq	master	Signal	PPU CME interrupt
ppu_core_irq	master	Signal	PPU core interrupt.
ppu_core_isolate	master	Signal	PPU Core Domain isolation control.
ppu_core_wakerequest	slave	Signal	PPU core wake request signal.
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	master	PVBus	The core will generate bus requests on this port.
pvbus_periph_m	master	PVBus	The core can generate peripheral bus request on this port.
rei	slave	Signal	Per core RAM Error Interrupt.
reset	slave	Signal	Reset signal to cluster.
rvbaraddr	slave	Value_64	Reset vector base address.
sci_m	master	SystemCoherencyInterface	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei	slave	Signal	Per core System Error physical pins.
spiden	slave	Signal	External debug interface.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq	master	Signal	Interrupt signal from the trace buffer unit.
utility_bus_s	slave	PVBus	Utility bus subordinate
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfiq	slave	Signal	Virtualised FIQ.
virq	slave	Signal	Virtualised IRQ.

Port	Direction	Protocol	Description
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Per core virtual System Error physical pins.

## Parameters for AEMvA\_DSUCT

### **cpuX.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **cpuX.CONFIG64**

Register width configuration at reset. 0, AArch32. 1, AArch64.

Type: `bool`

Default value: `true`

### **cpuX.CP15SDISABLE**

Initialize to disable access to some CP15 registers.

Type: `bool`

Default value: `false`

### **cpuX.CP15SDISABLE2**

Initialize to disable access to some CP15 registers (FEAT\_CP15SDISABLE2).

Type: `bool`

Default value: `false`

### **cpuX.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **cpuX.DCZID-log2-block-size**

Log2 of the block size in words cleared by DC ZVA instruction (as read from DCZID\_ELO).

Type: `uint8_t`

Default value: 8

### **`cpuX.DCZVA_single_write`**

Execute the DCZVA as a single write.

Type: `bool`

Default value: `false`

### **`cpuX.MPIDR-override`**

Override of MPIDR value. If nonzero will override the MT, cluster and CPU ID bits in MPIDR.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.RVBAR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.RVBAR32`**

Reset vector address in AARCH32 when VINITHI is not set and `ignore_rvbar_in_aarch32` is set.

Type: `uint32_t`

Default value: `0x0`

### **`cpuX.SMPnAMP`**

Enable broadcast messages necessary for correct SMP operation at reset.

Type: `bool`

Default value: `true`

### **`cpuX.TEINIT`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **`cpuX.VINITHI`**

Reset value of SCTLR.V.



Type: `bool`

Default value: `false`

### **`cpuX.aarch32_reset_from_impdef_addr`**

If PE resets into AArch32, Whether execution starts from IMPDEF address or hi/low vector.

Type: `bool`

Default value: `true`

### **`cpuX.ase-present`**

Set whether the model has been built with NEON support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`cpuX.clock_divider`**

Clock divider ratio for asymmetric MP clocking.

Type: `uint32_t`

Default value: `0x1`

### **`cpuX.clock_multiplier`**

Clock divider ratio for asymmetric MP clocking.

Type: `uint32_t`

Default value: `0x1`

### **`cpuX.crypto_aes`**

AES instructions supported (requires CryptoPlugin to be loaded). 0, not implemented. 2, AES and PMULL instructions implemented (FEAT\_AES, FEAT\_PMULL).

Type: `uint8_t`

Default value: `2`

### **`cpuX.crypto_sha1`**

SHA-1 instructions supported (requires CryptoPlugin to be loaded). 0, not implemented. 1, SHA1 instructions implemented (FEAT\_SHA1).

Type: `uint8_t`

Default value: `1`

**cpuX.crypto\_sha256**

SHA-256 instructions supported (requires CryptoPlugin to be loaded). 0, not implemented. 1, SHA256 instructions implemented (FEAT\_SHA256).

Type: uint8\_t

Default value: 1

**cpuX.crypto\_sha3**

Implement ARMv8.4 SHA-3 instructions (requires CryptoPlugin to be loaded) (FEAT\_SHA3).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

**cpuX.crypto\_sha512**

Implement ARMv8.4 SHA-512 instructions (requires CryptoPlugin to be loaded) (FEAT\_SHA512).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

**cpuX.crypto\_sm3**

Implement ARMv8.4 SM-3 instructions (requires CryptoPlugin to be loaded) (FEAT\_SM3).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

**cpuX.crypto\_sm4**

Implement ARMv8.4 SM-4 instructions (requires CryptoPlugin to be loaded) (FEAT\_SM4).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

**cpuX.cti-intack\_mask**

Set bits represent that the corresponding trigger requires software acknowledge via CTIINTACK.

Type: uint8\_t

Default value: 1

**cpuX.cti-number\_of\_claim\_bits**

Number of implemented bits in CTICLAIMSET.

Type: `uint8_t`

Default value: 0

**cpuX.cti-number\_of\_triggers**

Number of cti event triggers (default: 8, valid values: {3-32}).

Type: `uint8_t`

Default value: 8

**cpuX.enable\_crc32**

CRC32 instructions supported. 0, not implemented. 1, CRC32 instructions implemented (FEAT\_CRC32).

Type: `uint8_t`

Default value: 0

**cpuX.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**cpuX.etm-present**

Set whether the model has ETM support.

Type: `bool`

Default value: `true`

**cpuX.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `false`

**cpuX.force-fpsid-value**

Value to override the FPSID value to.

Type: `uint32_t`

Default value: `0x0`

### **`cpuX.has_hcptr_tase`**

If false, HCPTR.TASE is **RES0**.

Type: `bool`

Default value: `true`

### **`cpuX.highest-index-of-context-breakpoints`**

Highest index of breakpoints that are context aware.

Type: `uint8_t`

Default value: `15`

### **`cpuX.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.l2cache-read_bus_width_in_bytes`**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: `uint32_t`

Default value: `0x8`

### **`cpuX.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

### **`cpuX.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.l2cache-ways`**

L2 Cache number of ways (sets are implicit from size).

Type: `uint8_t`

Default value: 16

### **cpuX.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **cpuX.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x8

### **cpuX.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **cpuX.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

### **cpuX.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

### **cpuX.number-of-breakpoints**

Number of breakpoints.

Type: uint8\_t

Default value: 16

**cpuX.number-of-context-breakpoints**

Number of breakpoints that are context aware.

Type: `uint8_t`

Default value: 16

**cpuX.number-of-watchpoints**

Number of watchpoints.

Type: `uint8_t`

Default value: 16

**cpuX.operation\_bandwidth**

Operation width for ARMv8.4 PMU extension.

Type: `uint8_t`

Default value: 1

**cpuX.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**cpuX.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**cpuX.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: 0x123456

**cpuX.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**cpuX.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**cpuX.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**cpuX.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**cpuX.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**cpuX.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**cpuX.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**cpuX.semihosting-prefix**

Prefix semihosting output with target instance name.

Type: `bool`

Default value: `false`



**cpuX.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**cpuX.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**cpuX.semihosting-stderr\_istty**

Result for semihost istty call when argument is stderr.

Type: `bool`

Default value: `true`

**cpuX.semihosting-stdin\_istty**

Result for semihost istty call when argument is stdin.

Type: `bool`

Default value: `true`

**cpuX.semihosting-stdout\_istty**

Result for semihost istty call when argument is stdout.

Type: `bool`

Default value: `true`

**cpuX.semihosting-use\_stderr**

Send stderr from the simulated process to host stderr.

Type: `bool`

Default value: `false`

**cpuX.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`cpuX.unpredictable_WPMASKANDBAS`**

Constrained unpredictable handling of watchpoints when mask and BAS fields specified. 0, IGNOREMASK. 1, IGNOREBAS (default). 2, REPEATBAS8. 3, REPEATBAS.

Type: `uint8_t`

Default value: `1`

### **`cpuX.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`cpuX.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`cpuX.vfp-traps`**

Implement support for trapping floating-point exceptions.

Type: `bool`

Default value: `true`

### **`cpuX.vfp-traps-show-all`**

Report all trapped floating-point exceptions in the syndrome when a combination occurs.

Type: `bool`

Default value: `false`

### **`cpuX.wfet_early_or_delayed_timeout`**

WFET early or delayed timeout beyond the threshold value of CNTVCT\_ELO in percentage.

Type: `int8_t`

Default value: `0`

**cpuX.wfit\_early\_or\_delayed\_timeout**

WFIT early or delayed timeout beyond the threshold value of CNTVCT\_ELO in percentage.

Type: `int8_t`

Default value: 0

**ADFSR-AIFSR-implemented**

ADFSR and AIFSR are implemented.

Type: `bool`

Default value: `false`

**AEND0\_DEFAULT**

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

Type: `uint64_t`

Default value: `0x0`

**AEND1\_DEFAULT**

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

Type: `uint64_t`

Default value: `0x0`

**AEND2\_DEFAULT**

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

Type: `uint64_t`

Default value: `0x0`

**AEND3\_DEFAULT**

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

Type: `uint64_t`

Default value: `0x0`

**AIDR**

Value of AIDR\_EL1 register.

Type: `uint64_t`

Default value: `0x0`

**AMIIDR**

Value of AMU Implementation Identification Register.

Type: `uint64_t`

Default value: `0x43b`

**AMPIDR**

Value of AMU Peripheral Identification Register.

Type: `uint64_t`

Default value: `0x4000bb000`

**ASTART0\_DEFAULT**

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART1\_DEFAULT**

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART2\_DEFAULT**

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART3\_DEFAULT**

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

Type: `uint64_t`

Default value: `0x0`

**BPIMVA\_causes\_translation\_lookup**

Do a translation when BPIMVA instruction is executed (which may cause a translation fault).

Type: `bool`

Default value: `false`

**BROADCASTATOMIC**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`

Default value: `true`

**BROADCASTCACHEMAINT**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

Type: `bool`

Default value: `true`

**BROADCASTINNER**

Enable broadcasting of Inner Shareable transactions. The broadcastinner signal will override this value if used.

Type: `bool`

Default value: `true`

**BROADCASTOUTER**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

Type: `bool`

Default value: `true`

**BROADCASTPERSIST**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: `bool`

Default value: `true`

**CCSIDR-L1D\_override**

If nonzero, override the value presented in CCSIDR for L1D (this is cosmetic and does not affect cache behaviour).

Type: `uint64_t`

Default value: `0x0`

**CCSIDR-L1I\_override**

If nonzero, override the value presented in CCSIDR for L1I (this is cosmetic and does not affect cache behaviour).

Type: `uint64_t`

Default value: `0x0`

**CCSIDR-L2\_override**

If nonzero, override the value presented in CCSIDR for L2 (this is cosmetic and does not affect cache behaviour).

Type: `uint64_t`

Default value: `0x0`

**CCSIDR-L3\_override**

If nonzero, allow L3 selection in CSSELR and present this value in CCSIDR (this is cosmetic and does not affect cache behaviour).

Type: `uint64_t`

Default value: `0x0`

**CLUSTER\_ID**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the ManagerID64 of the core.

Type: `uint32_t`

Default value: `0x0`

### **CMO\_broadcast\_when\_cache\_state\_modelling\_disabled**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. This value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required.

Type: `uint8_t`

Default value: `1`

### **CPUCFR**

Value of CPU Configuration Register.

Type: `uint64_t`

Default value: `0x0`

### **CTIPIDR**

If non-zero, override the CTI Peripheral Identification Register.

Type: `uint64_t`

Default value: `0x0`

### **CTR-L1Ip-override**

If non-zero, override the L1Ip bits in CTR/CTR\_ELO system register. This does not change the behaviour of the cache, only what is present in the CTR register.

Type: `uint8_t`

Default value: `0`

### **DBGBCR\_BT\_applies\_RES0\_before\_valid\_check**

If true, **RES0** behaviour is applied to DBGBCR(EL1).BT before checking for reserved values for this field.

Type: `bool`

Default value: `true`

### **DBGPIDR**

If non-zero, override the Debug Peripheral Identification Register.

Type: `uint64_t`

Default value: `0x0`

#### **DBGROMADDR**

Initialization value of DBGDRAR register. Bits[55:12] of this register specify the ROM table physical address.

Type: `uint64_t`

Default value: `0x0`

#### **DBGROMADDRV**

If true, set bits[1:0] of the CP15 DBGDRAR to indicate that the address is valid.

Type: `bool`

Default value: `false`

#### **DCZID-TBS-log2-allocation-tags-block-size**

Log2 of the block size in words written by a DC ZGBVA or DC GBVA instruction (for FEAT\_MTETC).

Type: `uint8_t`

Default value: `9`

#### **DSU.cme.AMIIDR**

Value of AMU Implementation Identification Register.

Type: `uint64_t`

Default value: `0x43b`

#### **DSU.cme.AMPIDR**

Value of AMU Peripheral Identification Register.

Type: `uint64_t`

Default value: `0x400bbb000`

#### **DSU.cme.CMECFR**

Value of CMECFR\_EL1 fields (ECC, CHI) for all CMEs.

Type: `string`

Default value: `"{\\"ECC\\": 0, \\"CHI\\": 0}"`



**DSU.cme.amu\_aux\_type\_fixed**

Lists which AMU auxiliary registers that are fixed and to which event type. The JSON schema is: {fixed\_aux\_reg:evt\_type, ...}. For example {"0":0x300} would make auxiliary register 0 fixed to event type 0x300.

Type: string

Default value: N/A

**DSU.cme.amu\_aux\_voffset\_mask**

If ARMv8.6 is implemented, each bit of the field, 0 to 15, when 1 indicates that the corresponding virtual offset register, AMEVCNTVOFF1<n>\_EL2, is implemented.

Type: uint16\_t

Default value: 0x0

**DSU.cme.amu\_has\_external\_interface**

Implement external memory-mapped access to system register of activity monitor unit from ARMv8.4.

Type: bool

Default value: false

**DSU.cme.amu\_num\_auxiliary\_counters**

Number of AMU auxiliary counters implemented.

Type: uint8\_t

Default value: 0

**DSU.cme.amu\_reset\_domain**

Reset domain for activity monitor unit. 0, COLD\_RESET. 1, WARM\_RESET. 2, NONE.

Type: uint8\_t

Default value: 0

**DSU.cme.amu\_version**

Selects the activity monitor version implemented - 1, AMUv1 for Armv8.4 is implemented.- 2, AMUv1 for Armv8.6 is implemented (FEAT\_AMUv1p1).

Type: uint8\_t

Default value: 1

**DSU.cme.has\_amu**

Implement activity monitor functionality from ARMv8.4 (FEAT\_AMUv1).

Type: `bool`

Default value: `false`

**DSU.cme.mpmm\_accumulator\_multiplier**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator. value of `n` means the accumulator will use  $(n * \text{accumulator value})$  to calculate the mpmm threshold (MPMM). is provided as a fast model workaround to handle cases where execution of quantum in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

Type: `uint8_t`

Default value: `1`

**DSU.cme.power\_on\_by\_default**

If true, CME PPU's will initialize in Dynamic mode out of reset, enabling the CME to power itself on/off automatically as it is used.

Type: `bool`

Default value: `true`

**ERRIIDR**

Value of RAS Implementation Identification Register.

Type: `uint64_t`

Default value: `0xd800143b`

**ERRPIDR**

Value of RAS Peripheral Identification Register.

Type: `uint64_t`

Default value: `0x4100bbd80`

**ERXMISC0\_mask**

Write Mask for ERXMISC0 RAS Register.

Type: `uint64_t`

Default value: `0x0`

**GICDISABLE**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`

Default value: `true`

**GMID-log2-block-size**

Log2 of the block size in words accessed by STGM/LDGM/STZGM instructions.

Type: `uint8_t`

Default value: `4`

**ISV\_set\_to\_0\_for\_page\_boundary\_crossing**

Whether ESR\_ELx.ISV is cleared when a faulted access crosses a page boundary and FAR\_ELx reports the subsequent page.

Type: `bool`

Default value: `false`

**ISV\_set\_to\_0\_for\_stage2\_synch\_external\_abort**

Whether ESR\_EL2.ISV is set to 0 on stage 2 synchronous external aborts.

Type: `bool`

Default value: `false`

**MIDR**

Value of MIDR\_EL1 register.

Type: `uint32_t`

Default value: `0x410fd0f0`

**NUM\_CMES**

Defines how many CMEs are associated with each cluster. This parameter must be in sync with SVE.ScalableVectorExtension SME configuration.

Type: `uint8_t`

Default value: `0`

**NUM\_CORES**

Number of cores per cluster.

Type: `uint8_t`

Default value: 1

### **PA\_SIZE**

Physical address range supported For ARMv8.0 and ARMv8.1 this is limited to 48 bits (FEAT\_LPA).

Type: `uint8_t`

Default value: 40

### **PERIPHBASE**

Base address of peripheral memory space.

Type: `uint64_t`

Default value: 0x13080000

### **PMCEID0**

Performance Monitor Common Event ID Reg 0 value - 64 bit.

Type: `uint64_t`

Default value: 0xffffffff

### **PMCEID1**

Performance Monitor Common Event ID Reg 1 value - 64 bit.

Type: `uint64_t`

Default value: 0xffffffff

### **PMSIDR.ArchInst**

Defines whether architecture instruction sampling is implemented or not, if not only micro op sampling is implemented. Model only supports architecture instruction sampling, but allows ID register field to be configured.

Type: `bool`

Default value: `true`

### **PMSIDR.CRR**

Defines whether call return branch records (FEAT\_SPE\_CRR) is implemented or not.

Type: `bool`

Default value: `false`

**PMSIDR.LDS**

Defines whether data source for sampled load instruction is implemented or not. Model does not implement loaded data source, but allows ID register field to be configured.

Type: `bool`

Default value: `false`

**PMUPIDR**

If non-zero, override the PMU Peripheral Identification Register.

Type: `uint64_t`

Default value: `0x0`

**VAL\_disable\_slpoe2\_scr\_el3\_traps**

Disable traps from SCR\_EL3 for FEAT\_S1POE2, FEAT\_TPS and FEAT\_TPSP. WARNING: this is a temporary parameter to aid with software support for these traps. It will be removed in the future.

Type: `bool`

Default value: `false`

**abort\_execution\_from\_device\_memory**

Execution from device memory generates a prefetch abort.

Type: `bool`

Default value: `false`

**abort\_slpoe2\_fetch\_from\_device\_memory**

Abort S1POE2 table fetches from device memory.

Type: `bool`

Default value: `false`

**acp\_width\_in\_bits**

ACP port width in bits. 0 when no ACP port is configured. 128 and 256 are allowed when ACP port is configured.

Type: `uint16_t`

Default value: `0`

**advsimd\_bf16\_support\_level**

Implement BFloat16 operations from ARMv8.6. AArch64 Advanced SIMD and FP BFloat16 instructions are automatically enabled when `has_arm_v8-6` is true.- 0, Not implemented.- 1, AArch64 Advanced SIMD and FP BFloat16 instructions only (FEAT\_BF16).- 2, AArch32 Advanced SIMD and VFP BFloat16 instructions only (FEAT\_AA32BF16).- 3, Both AArch64 Advanced SIMD and FP and AArch32 Advanced SIMD and VFP BFloat16 instructions.

Type: `uint8_t`

Default value: 0

**advsimd\_i8mm\_support\_level**

Implement Int8 matrix multiply operations from ARMv8.6. AArch64 Advanced SIMD and FP Int8 matrix multiply instructions are automatically enabled when `has_arm_v8-6` is true.- 0, Not implemented.- 1, AArch64 Advanced SIMD and FP Int8 matrix multiply instructions only (FEAT\_I8MM).- 2, AArch32 Advanced SIMD and VFP Int8 matrix multiply instructions only (FEAT\_AA32I8MM).- 3, Both AArch64 Advanced SIMD and FP and AArch32 Advanced SIMD and VFP Int8 matrix multiply instructions (FEAT\_I8MM, FEAT\_AA32I8MM).

Type: `uint8_t`

Default value: 0

**advsimd\_overread**

AdvSIMD element load operations access all bytes of a 16-byte aligned window, even in Device memory.

Type: `bool`

Default value: `false`

**align\_pc\_on\_branch\_to\_unaligned\_pc\_aarch32**

Force PC align for branches to an unaligned PC counter in A32 state.

Type: `bool`

Default value: `false`

**align\_pc\_on\_debug\_exit\_to\_aarch32**

Exit to AARCH32 state from debug state forces pc bit0 to 0.

Type: `bool`

Default value: `false`

**align\_pc\_on\_illegal\_exception\_return\_to\_aarch32**

Align PC when performing an illegal exception return from AArch64 to AArch32.

Type: `bool`

Default value: `true`

### **`allow_s1_dbm_update_on_s2_mmu_fault`**

Whether s1 dirty bit update is done when s2 of ipa (not s1 ttw) generates mmu fault.

Type: `bool`

Default value: `true`

### **`amair_reg_rw_mask`**

RW mask for implementation-defined registers.

Type: `uint64_t`

Default value: `0x0`

### **`amu_aux_counter_mask`**

If ARMv8.4 is implemented, each bit of the field, 0 to 15, when 1 indicates that the corresponding virtual offset register, AMEVCNTR1<n>\_ELO/AMEVTYPER1<n>\_ELO, is implemented.

Type: `uint16_t`

Default value: `0x0`

### **`amu_aux_type_fixed`**

Lists which AMU auxiliary registers that are fixed and to which event type. The JSON schema is: {fixed\_aux\_reg:evt\_type, ...}. For example {"0":0x300} would make auxiliary register 0 fixed to event type 0x300.

Type: `string`

Default value: `N/A`

### **`amu_aux_voffset_mask`**

If ARMv8.6 is implemented, each bit of the field, 0 to 15, when 1 indicates that the corresponding virtual offset register, AMEVCNTVOFF1<n>\_EL2, is implemented.

Type: `uint16_t`

Default value: `0x0`

### **`amu_has_external_interface`**

Implement external memory-mapped access to system register of activity monitor unit from ARMv8.4 (FEAT\_AMU\_EXT).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **`amu_has_sysreg_interface`**

Implement system register access to activity monitor unit from ARMv8.4.

Type: `bool`

Default value: `true`

### **`amu_mmap_address`**

AMU base address for each core on system bus. 0 means the AMU is not mapped, otherwise the address must be 4KB aligned. JSON schema for the parameter value is: `{"format": "all_addrs_are_absolute_wrt_systembus", "cores": [{"amu": 0x0}, {"amu": 0x0}, {"amu": 0x0}, {"amu": 0x0}]}`.

Type: `string`

Default value: N/A

### **`amu_num_auxiliary_counters`**

Number of AMU auxiliary counters implemented.

Type: `uint8_t`

Default value: 0

### **`amu_reset_domain`**

Reset domain for activity monitor unit. 0, COLD\_RESET. 1, WARM\_RESET. 2, NONE.

Type: `uint8_t`

Default value: 0

### **`amu_version`**

Selects the activity monitor version implemented - 1, AMUv1 for Armv8.4 is implemented.- 2, AMUv1 for Armv8.6 is implemented (FEAT\_AMUv1p1).

Type: `uint8_t`

Default value: 1

### **`apsr_read_restrict`**

At ELO, unknown bits of APSR are **RAZ**.

Type: `bool`



Default value: `false`

### **arm\_v8\_7\_accelerator\_support\_level**

Implements accelerator support instructions: 0, Not implemented 1, FEAT\_LS64 implemented 2, FEAT\_LS64\_V implemented 3, FEAT\_LS64\_ACCDATA implemented 4, FEAT\_LS64WB implemented.

Type: `uint8_t`

Default value: 0

### **atomic\_memtype\_fault\_prio\_less\_than\_gpc\_fault**

Bitmask controlling whether unsupported memtype faults are lower priority than GPC faults: bit0 applies to atomic/exclusive accesses, bit1 applies to LS64.

Type: `uint8_t`

Default value: 0

### **atomic\_memtype\_fault\_priority**

This parameter describes the priority of unsupported atomic/exclusive memtype fault w.r.t alignment and permission fault. 0, BEFORE\_ALIGN\_MEM\_FAULT. 1, AFTER\_ALIGN\_BEFORE\_PERM\_FAULT. 2, AFTER\_PERM\_FAULT.

Type: `uint8_t`

Default value: 0

### **auxilliary\_feature\_register0**

Value of AFRO ID register.

Type: `uint32_t`

Default value: 0x0

### **branch-predictor-clear-policy**

Set branch prediction policy as defined for MMFR1[31:28]. This does not change the behaviour of the branch predictor, only what is reported in MMFR1.BPred.

Type: `uint8_t`

Default value: 2

### **branch-predictor-supported-ops**

Set branch prediction policy as defined for MMFR3[11:8]. This does not change the behaviour of the branch predictor, only what is reported in MMFR3.BPMaint.

Type: `uint8_t`

Default value: 1

### **brbe\_disable\_recording**

If BRBE is implemented and this is set to true, disable BRBE recording. All registers will be functional, but no branches will be recorded. This will improve model performance for workloads that enable BRBE, but don't care about the information stored in it. (FEAT\_BRBE).

Type: `bool`

Default value: `false`

### **brbe\_log2\_num\_records**

Log2 of number of BRB records supported. 3 -> 8 records, ... 6 -> 64 records.

Type: `uint8_t`

Default value: 6

### **brbinf\_type\_override\_on\_impdef\_trap\_to\_el3**

If true, force BRBINF.TYPE=0x23 (trap) when ESR.EC=0x1f (implementation defined exception to EL3) (FEAT\_BRBE).

Type: `bool`

Default value: `false`

### **bti\_support\_level**

Support branch target identification: 0 - None. 1 - Support FEAT\_BTI (mandatory from ARMv8.5). 2 - Support FEAT\_BTIE.

Type: `uint8_t`

Default value: 0

### **cache-log2linelen**

Log2 of the cache line length in bytes.

Type: `uint8_t`

Default value: 6

### **cache\_maintenance\_hits\_watchpoints**

DCIMVA operations executed in AArch32 modes hit watchpoints.

Type: `bool`

Default value: `false`

**cache\_mlb\_with\_default\_id**

If true, incomplete mlb fetches that ends up with entries with default MPAM ids will still be cached.

Type: `bool`

Default value: `false`

**changing\_block\_size\_without\_bbm\_support**

Changing block size without break-before-make support level(OPTIONAL from Armv8.3): 0, Unsupported, 1, Level 1 support (FEAT\_BBML1), 2, Level 2 support (FEAT\_BBML2), 3, Level 3 support (FEAT\_BBML3).

Type: `uint8_t`

Default value: 0

**check\_memory\_attributes**

Detect and report TLB use of conflicting memory attributes for views of the same physical address.

Type: `bool`

Default value: `false`

**checked\_pointer\_arithmetic\_support\_level**

Specify the Checked Pointer Arithmetic support level: 0, not implemented. 1, FEAT\_CPA is implemented. 2, FEAT\_CPA2 is implemented.

Type: `uint8_t`

Default value: 0

**clean\_invalidate\_cache\_on\_warm\_reset**

Clean and invalidate caches on warm reset.

Type: `bool`

Default value: `false`

**clear\_IT\_when\_IL\_set**

Clear IT bits when performing a *legal* exception return to AArch32 when IL is set.

Type: `bool`

Default value: `false`

**clear\_IT\_when\_IL\_set\_explicitly**

Apart from `clear_IT_when_IL_set`, also clear IT bits when loading CPSR from SPSR/memory and `IL == 1` in the value being loaded.

Type: `bool`

Default value: `false`

**clear\_ec\_in\_debug\_state**

When ARMv8.8 debug extension is implemented, whether EDESR.EC bit is set/cleared on entering debug state due to pending exception catch caused by EDESR.EC=1.

Type: `bool`

Default value: `false`

**clear\_reg\_top\_eret**

Behaviour of the upper 32-bits of the Xn registers when changing between AArch32 state and AArch64 state. 0, upper 32-bits preserved for all registers. 1, upper 32-bits set to 0 for all accessible registers. 2, upper 32-bits set to 0 for a random selection of accessible registers. 3, upper-32-bits set to 0 for registers touched in AArch32.

Type: `uint8_t`

Default value: 1

**clear\_reg\_top\_set**

Whether to clear upper 32-bits of the Xn register when corresponding AArch32 register is set via Iris.

Type: `bool`

Default value: `true`

**cluster\_patch\_level**

Cosmetic change to patch number in CLUSTERIDR. Corresponds to the Y in rXpY.

Type: `uint8_t`

Default value: 0

**cluster\_pmu-num\_counters**

Number of cluster-level pmu counters implemented.

Type: `uint8_t`

Default value: 6

**cluster\_revision\_number**

Cosmetic change to revision number in CLUSTERIDR, Corresponds to the X in rXpY.

Type: `uint8_t`

Default value: 0

**cluster\_split\_lock\_config**

Default SPLIT/LOCKED config. Directly maps to values of CLUSTERSLCFR. The valid values are: 1 - Only LOCKED configuration support 4 - Only SPLIT configuration support, 5 - Mixed Configuration support. Modes are not software visible, and not modeled. Valid only when `enable_ae_features` is true.

Type: `uint8_t`

Default value: 1

**configure\_pmu\_events\_with\_json**

"Configure v8.6 and newer PMU events. Note : This param has high priority and overrides the setting of "has\_\*\_pmu\_events" if both the params are provided. JSON schema for the parameter value is e.g. 1. {"all":false} 2. {"pmu\_events":["EVENT\_NAME\_1","EVENT\_NAME\_2"]}."

Type: `string`

Default value: N/A

**configure\_v8\_6\_pmu\_events\_with\_json**

"[DEPRECATED: Set `configure_pmu_events_with_json` to the same value instead] Configure v8.6 PMU events. Note : This param has high priority and overrides the setting of "has\_v8\_6\_pmu\_events" if both the params are provided. JSON schema for the parameter value is e.g. 1. {"all":false} 2. {"pmu\_events":["BR\_INDNR\_RETIRE", "BR\_IND\_RETIRE", "BR\_RETURN\_SKIP\_RETIRE", "BR\_RETURN\_ANY\_RETIRE", "BR\_INDNR\_SKIP\_RETIRE", "BR\_INDNR\_TAKEN\_RETIRE", "BR\_IND\_SKIP\_RETIRE", "BR\_IND\_TAKEN\_RETIRE", "BR\_IMMED\_SKIP\_RETIRE", "BR\_IMMED\_TAKEN\_RETIRE", "BR\_SKIP\_RETIRE"]}."

Type: `string`

Default value: N/A

**configure\_v8\_8\_pmu\_events\_with\_json**

"[DEPRECATED: Set `configure_pmu_events_with_json` to the same value instead] Configure v8.8 PMU events. Note : This param has high priority and overrides the setting of "has\_v8\_8\_pmu\_events" if both the params are provided. JSON schema for the parameter value is e.g. 1. {"all":false} 2. {"pmu\_events":["BR\_HINT\_COND\_RETIRE", "BR\_COND\_TAKEN\_RETIRE", "BR\_UNCOND\_RETIRE", "BR\_COND\_RETIRE", "BRNL\_TAKEN\_RETIRE", "BRNL\_IND\_TAKEN\_RETIRE", "BRNL\_INDNR\_TAKEN\_RETIRE"]}."

"BRNL\_IMMED\_TAKEN\_RETIRED", "BL\_TAKEN\_RETIRED", "BL\_IND\_TAKEN\_RETIRED",  
"BL\_IMMED\_TAKEN\_RETIRED"]}]".

Type: `string`

Default value: `N/A`

### **configure\_v8\_9\_pmu\_events\_with\_json**

"[DEPRECATED: Set `configure_pmu_events_with_json` to the same value instead] Configure v8.9 PMU events. Note : This param has high priority and overrides the setting of `"has_v8_9_pmu_events"` if both the params are provided. JSON schema for the parameter value is e.g. 1. `{"all":false}` 2. `{"pmu_events":["ASE_SVE_RETIRED", "ASE_RETIRED", "VFP_RETIRED", "SVE_RETIRED", "CRYPTO_RETIRED", "SIMD_INST_RETIRED", "ASE_INST_RETIRED", "SVE_INST_RETIRED", "ASE_SVE_INST_RETIRED", "LD_ANY_RETIRED", "ST_ANY_RETIRED", "LDST_ANY_RETIRED", "DP_RETIRED"]}]`".

Type: `string`

Default value: `N/A`

### **core\_cache\_protection**

`core_cache_protection` can change `ERROFR`, `ERROPFGF` and `ERROPFGCTL` fields. Possible values are: -1:Not implemented (by default), 0:Disabled, 1:Enabled.

Type: `int8_t`

Default value: `-1`

### **core\_complex\_mapping**

Defines Complex descriptions for platforms that support several Cores per Complex like Cortex-A510. JSON format: `{"complex0": { "cores" : [ 0, 1 ], "l2-cache" :{"exists":1, "size":16MB}}, ... , "complexN": { "cores" : [<core_list>], "l2-cache" : {"exists":1, "size":16MB}}}` where `<core_list>` is the list of cores in the complexN. Effective only when the parameter value is not empty.

Type: `string`

Default value: `N/A`

### **core\_power\_on\_by\_default**

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPU which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

Type: `bool`

Default value: `false`

**cpacr\_trcdis\_behaviour**

Behaviour of CPACR.TRCDIS/NSACR.NSTRCDIS when there is no CP14 ETM interface. 0, **RAZ/WI**. 2, implemented.

Type: `uint8_t`

Default value: 2

**cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: 0x1

**cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: 0x1

**cpuselr\_el3\_sync\_immediate**

Adjust when the patching selection register synchronises - either immediately (true - default), or awaiting for barrier event.

Type: `bool`

Default value: `true`

**cpy\_mops\_option**

Set option for Armv8.8 CPY(FEAT\_MOPS). 0, use default(i.e. use value configured through `has_mops_option`). 1, implemented using Option A. 2, implemented using Option B.

Type: `uint8_t`

Default value: 0

**cpyf\_mops\_option**

Set option for Armv8.8 CPYF(FEAT\_MOPS). 0, use default(i.e. use value configured through `has_mops_option`). 1, implemented using Option A. 2, implemented using Option B.

Type: `uint8_t`

Default value: 0

**cycle\_counter\_freeze\_on\_spe\_event**

If true, pmu cycle counter does not count when pmcr\_el0.dp=1 and pmu event counters are frozen by pmcr\_el0.fzs. (FEAT\_SPE\_DPFZS).

Type: bool

Default value: false

**d128\_disabled\_ps\_resvd\_size**

Physical size treated when TCR.(I)PS is programmed with value seven and D128 is disabled via TCR. 0, 48 bits. 1, 52 bits. 2, 56.

Type: uint8\_t

Default value: 2

**data\_abort\_on\_gcs\_access\_to\_non\_normal\_memory**

If true, a GCS data access to non-normal memory results in a data abort for unsupported access.

Type: bool

Default value: false

**dbg-bcr-reserved-behavior**

This is the behavior of the reserved values of the BT field in DBGBCR. Possible values are: - 0 = Disabled. - 1 = BT[2] is ignored..

Type: uint8\_t

Default value: 1

**dbg\_rom\_dap\_addr**

Debug ROM dap base address.

Type: uint64\_t

Default value: 0x0

**dbgitr\_buffer\_size**

Number of instructions which can be buffered before EDSCR.ITE is cleared.

Type: uint32\_t

Default value: 0x0



**dbgxvr\_ress\_is\_stateful**

Whether DBGWVR/DBGBVR.RESS returns last written value. if set to false, RESS returns sign extended value.

Type: `bool`

Default value: `false`

**dc\_fault\_unaligned\_s1\_device\_s2\_fwb**

Whether takes an Alignment Fault caused by the memory type on a DC {ZVA,GZVA,GVA} if the stage 1 memory type is any Device memory type.

Type: `bool`

Default value: `false`

**dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-maintenance\_latency**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-miss\_latency**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-prefetch\_enabled**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

Type: `bool`

Default value: `false`

**dcache-read\_access\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**dcache-read\_bus\_width\_in\_bytes**

L1 D-Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x8

**dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**dcache-size**

L1 D-Cache size in bytes.

Type: uint32\_t

Default value: 0x8000

**dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**dcache-state\_modelled**

Set whether D-cache has stateful implementation.

Type: bool

Default value: `false`

### **`dcache-ways`**

L1 D-Cache number of ways (sets are implicit from size).

Type: `uint8_t`

Default value: 2

### **`dcache-write_access_latency`**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`dcache-write_bus_width_in_bytes`**

L1 D-Cache write bus width in bytes used to calculate per-access timing annotations.

Type: `uint32_t`

Default value: `0x8`

### **`dcache-write_latency`**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`dcimva_requires_s2_write_permissions`**

Data-cache invalidate by MVA operations require stage 2 write permission (virtualised AArch32 guest).

Type: `bool`

Default value: `false`

### **`dczva_reports_lowest_addr_on_tag_check_fail`**

Whether DC ZVA reports lowest address in FAR on tag check fail.

Type: `bool`

Default value: `false`

### **dczva\_wp\_far\_behaviour**

Set option for address stored in FAR/EDWARD after watchpoints hit by DC ZVA. - FAR recorded matches lowest watchpointed address accessed by the instruction - FAR recorded matches lowest address accessed by the instruction within same translation granule as watchpointed address - FAR recorded matches highest address accessed by the instruction within same translation granule as watchpointed address.

Type: `uint8_t`

Default value: 0

### **debug\_auth\_signals\_sampled\_at\_reset**

Debug authentication signals can be configured as either sampled at reset only or at any time for External Root Debug.

Type: `bool`

Default value: `false`

### **debug\_components\_dap\_address**

Debug components ROM,ED,CTI,PMU,TRACE and TRBU base address for each core on debug bus. The "rom" field in the "cores" array are only allowed when 'debug\_rom\_is\_flat' is false. JSON schema for the parameter value is: {"format":"all\_addrs\_are\_absolute\_wrt\_debugbus", "cores": [{"rom":0x0, "ed":0x0, "cti":0x0, "pmu":0x0, "etm":0x0, "trbu":0x0}, {"rom":0x0, "ed":0x0, "cti":0x0, "pmu":0x0, "etm":0x0, "trbu":0x0}, {"rom":0x0, "ed":0x0, "cti":0x0, "pmu":0x0, "etm":0x0, "trbu":0x0}, {"rom":0x0, "ed":0x0, "cti":0x0, "pmu":0x0, "etm":0x0, "trbu":0x0}]}

Type: `string`

Default value: N/A

### **debug\_components\_mmap\_address**

Debug components ROM,ED,CTI,PMU,TRACE and TRBU base address for each core on system bus. The "rom" field in the "cores" array are only allowed when 'debug\_rom\_is\_flat' is false. JSON schema for the parameter value is: {"format":"all\_addrs\_are\_absolute\_wrt\_systembus", "cores": [{"rom":0x0, "ed":0x0, "cti":0x0, "pmu":0x0, "etm":0x0, "trbu":0x0}, {"rom":0x0, "ed":0x0, "cti":0x0, "pmu":0x0, "etm":0x0, "trbu":0x0}, {"rom":0x0, "ed":0x0, "cti":0x0, "pmu":0x0, "etm":0x0, "trbu":0x0}, {"rom":0x0, "ed":0x0, "cti":0x0, "pmu":0x0, "etm":0x0, "trbu":0x0}]}

Type: `string`

Default value: N/A

**debug\_entry\_is\_context\_sync**

If true, Entry in debug state is Context sync. Exiting debug state is a context synchronising operation, but entering is not. However some cpu implementation can consider also the Entry in Debug state as a CSE.

Type: `bool`

Default value: `false`

**debug\_rom\_is\_class\_9**

If true, present a debug ROM table as a class 9 device. Otherwise, use a class 1 ROM table.

Type: `bool`

Default value: `false`

**debug\_rom\_is\_flat**

If true, present a debug ROM table recommended by ARMv8 Debug Architecture. Otherwise, use nested ROM tables.

Type: `bool`

Default value: `false`

**default\_opmode**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

Type: `uint8_t`

Default value: 4

**delay\_serror**

Add a propagation delay of serror signal into the core.

Type: `uint32_t`

Default value: 0x0

**delayed\_dbgreg\_between\_secure\_views**

If `delayed_dbgreg` is enabled, whether the secure and nonsecure external views require explicit synchronization. values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

**delayed\_pmureg\_between\_secure\_views**

If `delayed_pmureg` is enabled, whether the secure and nonsecure external views require explicit synchronization. values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

**dfr1\_reads\_actual\_bp\_wp\_ctx\_cmp**

If true, the register `ID_AA64DFR1_EL1/EDDFR1` reports the actual number of BRPs, WRPs and CTX\_CMPs even when the number of `bp/wp/ctx_cmp` is less than 16.

Type: `bool`

Default value: `false`

**diagnostics**

Enable DynaMiq diagnostic messages.

Type: `bool`

Default value: `false`

**dic-spi\_count**

Number of shared peripheral interrupts implemented.

Type: `uint8_t`

Default value: 64

**disable\_bti\_effects**

When BTI effects are disabled: `PSTATE.BTYPE` is never updated and BTI exceptions are never triggered. This is not architecturally correct behaviour, but can lead to improved simulation performance.

Type: `bool`

Default value: `false`

**disable\_sve\_plugin**

If true, SVE will not be implemented in this processor even if the plugin is loaded (`FEAT_SVE`).

Type: `bool`

Default value: `false`

**disable\_unknown\_update\_event\_on\_reset**

Disables SYSREG\_UPDATE event notification on reset for the registers whose bitfields are all reserved or resets to architecturally unknown value.

Type: `bool`

Default value: `false`

**dsb\_accumulate\_threshold**

Limit the maximum number of observable DSB side effects which can be queued (e.g. TLBI), after which DSB sync will be done automatically.

Type: `uint32_t`

Default value: `0x100`

**e2h\_forces\_interrupt\_overrides**

If true, HCR\_EL2.xMO are treated as 1 else as programmed.

Type: `bool`

Default value: `false`

**early\_implicit\_error\_sync\_event\_behaviour**

Set option for Early Implicit Error Synchronization event (FEAT\_IESB) `0x0` - Behavior is not described ID\_AA64MMFR4\_EL1.EIESB = `0x00x1` - Implicit Error synchronization event is inserted before an exception is taken to EL3 (depending on SCR\_EL3.NMEA) ID\_AA64MMFR4\_EL1.EIESB = `0x10x2` - Implicit Error synchronization event is inserted before an exception is taken to ELx (depending on SCR\_EL3.NMEA and SCTLR2\_ELx.NMEA) ID\_AA64MMFR4\_EL1.EIESB = `0x20xF` - Implicit Error synchronization event is inserted after an exception is taken ID\_AA64MMFR4\_EL1.EIESB = `0xF`.

Type: `uint8_t`

Default value: `0`

**ecv\_support\_level**

Implement Enhanced Counter Virtualization feature from ARMv8.6. 0, Not supported. 1, fully supported without CNTPOFF. 2, fully supported with CNTPOFF (FEAT\_ECV).

Type: `uint8_t`

Default value: `0`

**eddfr1\_reads\_idreg\_mask**

Mask to configure each bitfield for EDDFR1 register, whether to be read from corresponding bitfield in AA64DFR1 register. Note: Only those bitfield which supports 'same as ID register or res0' functionality can be configured.

Type: `uint64_t`

Default value: `0xffffffffffffffff`

**eddfr2\_reads\_idreg\_mask**

Mask to configure each bitfield for EDDFR2 register, whether to be read from corresponding bitfield in ID\_AA64DFR2 register. Note: Only those bitfield which supports 'same as ID register or res0' functionality can be configured.

Type: `uint64_t`

Default value: `0xffffffffffffffff`

**eddfr\_reads\_idreg\_mask**

Mask to configure each bitfield for EDDFR register, whether to be read from corresponding bitfield in AA64DFR register. Note: Only those bitfield which supports 'same as ID register or res0' functionality can be configured.

Type: `uint64_t`

Default value: `0xffffffffffffffff`

**edpfr\_ras\_unknown\_bits\_read\_as\_0**

If true then **UNKNOWN** bits in RAS field in EDPFR are read as 0.

Type: `bool`

Default value: `false`

**edxfr\_reads\_idreg**

Whether EDDFR,EDDFR1 and EDDFR2 reads corresponding bitfield value from ID\_AA64DFR reg. Also, when this parameter is enabled, bitfields of these registers are configurable through 'eddfr\*\_reads\_idreg\_mask' parameter.

Type: `bool`

Default value: `false`

**el0\_can\_access\_imp\_def\_functionality**

If not made UNDEF by `imp_def_functionality_behaviour`, ELO can access **IMPLEMENTATION DEFINED** registers and system instructions.

Type: `bool`



Default value: `false`

### **`el0_el1_only_non_secure`**

Secure/non-secure state if EL2 and EL3 are not implemented. 0, secure. 1, non-secure.

Type: `bool`

Default value: `false`

### **`el3_trap_priority_when_secure_debug_disabled`**

Undef when secure debug is disabled (EDSCR.SDD == 1) && boolean IMPLEMENTATION\_DEFINED 'EL3 trap priority when SDD == 1'.

Type: `bool`

Default value: `false`

### **`enable-gicv5`**

if enable-gicv5 is set, then GICv5 is Supported.

Type: `bool`

Default value: `false`

### **`enable_address_contig_check`**

Check the input address range for the table entries that have the contiguous hint bit set.

Type: `bool`

Default value: `false`

### **`enable_ae_features`**

True when AE features are enabled for cluster, false otherwise.

Type: `bool`

Default value: `false`

### **`enable_debug_access_trace`**

If true, enables traces on debug access. Currently enables following traces on debug :: MMU\_TRANS.

Type: `bool`

Default value: `false`

**enable\_debug\_auth\_signals\_config**

Debug Authentication Signals DBGEN, SPIDEN (and if RME is enabled RLPIDEN and RTPIDEN) are configurable (default) or not configurable, (hardwired to 1). This parameter is the integer representation of a bitmap to enable configuration of these signals, with:- BIT[0] = DBGEN- BIT[1] = SPIDEN- BIT[2] = RLPIDEN- BIT[3] = RTPIDEN.

Type: `uint8_t`

Default value: 15

**enable\_mpam\_mvms\_mlb\_cache**

If true, MPAMv2\_VID MLB entries for MVMS (VPARTID/VPMG \342\206\222 MITT base) are always cached until invalidation, otherwise not cached.

Type: `bool`

Default value: `true`

**enable\_mpam\_vid\_pid\_mlb\_cache**

If true, MPAMv2\_VID MLB entries for VPARTID/VPMG \342\206\222 physical PARTID/PMG are always cached until invalidation, otherwise not cached.

Type: `bool`

Default value: `true`

**enable\_simulation\_performance\_optimizations**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of `stage12_tlb_size` parameter to 1024).

Type: `bool`

Default value: `true`

**enable\_tlb\_contig\_check**

Perform extra pagetable walks to check translation table entries that have the contiguous hint bit set. Ignored if FEAT\_BBML3 is implemented.

Type: `bool`

Default value: `false`

**enhanced\_pac2\_level**

Implements Enhanced PAC2 from ARMv8.6 (FEAT\_PAuth2), and PAC enhancements from ARMv9.5 (FEAT\_PAuth\_LR). options 0-3 of this feature are mandatory for ARMv8.6 but can be

cherry-picked to a ARMv8.3(or greater) implementation. FEAT\_FPACCOMBINE is mandatory in the presence of Future Architecture Technologies (FAT). 0: No EnhancedPAC2, 1: EnhancedPAC2 Only (FEAT\_PAuth2), 2: EnhancedPAC2 with FPAC (FEAT\_FPAC), 3: EnhancedPAC2 with FPACCombined (FEAT\_FPACCOMBINE), 4: EnhancedPAC2 with LR signing (FEAT\_PAuth\_LR).

Type: `uint8_t`

Default value: 0

### **error\_record\_feature\_register**

RAS feature register values. An array of JSON objects. The JSON schema for the array is: [{"ED":0x0, "IMPDEF\_3\_2":0x0, "UI":0x0, "FI":0x0, "UE":0x0, "CFI":0x0, "CEC":0x0, "RP":0x0, "DUI":0x0, "CEO":0x0, "CI":0x0, "TS":0x0, "INJ":0x0, "FRX":0x0, "UC":0x0, "UEU":0x0, "UER":0x0, "UEO":0x0, "DE":0x0, "CE":0x0, "Visibility":"Core"},other\_feature\_register\_values]. Where ED,UI,FI,CE and UE have valid values between 0x0 - 0x3. CFI and DUI have valid values 0x0, 0x2 and 0x3. CEC has valid values 0x0,0x2 or 0x4. RP,CEO,INJ,FRX,UC,UEU,UER,UEO,DE has valid values 0x0 or 0x1. CI and TS has valid values of 0x0, 0x1 and 0x2. Visibility has valid values "Core" or "Cluster".

Type: `string`

Default value: N/A

### **error\_record\_feature\_register\_json\_file**

File path to the RAS feature register values as JSON. The file uses the same format as the `error_record_feature_register` parameter value.

Type: `string`

Default value: N/A

### **erxpgctl\_res0\_stateful\_mask**

Mask for stateful bits for ERXPGCTL which are **RES0**.

Type: `uint64_t`

Default value: 0x0

### **esr\_write\_update\_res0**

If true, and RASv2 is enabled, then ESR\_ELx.WU field is **RES0** for errors on both loads and stores (FEAT\_RASv2).

Type: `bool`

Default value: `false`

**ete.ASYNC\_PACKETS\_WHEN\_VIEWINST\_OFF**

Generate the non-periodic alignment synchronisation packet generation when trace unit is operative.

Type: `bool`

Default value: `false`

**ete.ATBTRIG**

ATB trigger support.

Type: `bool`

Default value: `true`

**ete.CCITMIN**

Minimum cycle count value.

Type: `uint16_t`

Default value: `0x4`

**ete.CCSIZE**

Cycle counter size.

Type: `uint8_t`

Default value: `12`

**ete.CLAIMTAGS**

Number of claim tags.

Type: `uint8_t`

Default value: `8`

**ete.COMMOPT**

Commit mode.

Type: `bool`

Default value: `true`

**ete.DEBUG**

DEBUG.

Type: `uint8_t`

Default value: 2

**ete.DESIGNER**

DESIGNER value.

Type: `uint8_t`

Default value: 65

**ete.ETE\_REVISION**

ETE revision: 0=ETEv1.0, 1=ETEv1.1, 2=ETEv1.2, 3=ETEv1.3.

Type: `uint8_t`

Default value: 0

**ete.EXCEPTION\_WITH\_CONTEXT**

Whether EXCEPTION\_WITH\_CONTEXT packet is supported.

Type: `bool`

Default value: `true`

**ete.EXPLICITLY\_COMMIT\_PO\_ELEMS**

Whether to unilaterally explicitly emit a commit after a PO packet.

Type: `bool`

Default value: `false`

**ete.IMPDEFEXCEPPERCENTAGE**

Percentage of IMPDEF exceptions inserted in instruction blocks.

Type: `uint8_t`

Default value: 0

**ete.IMPDEF\_TRACE\_ON**

Whether trace is flushed and trace on packet generated by events described by bitmap value. bit 0 - PE entering low power state, bit 1 - PE entering debug state.

Type: `uint8_t`

Default value: 0

**ete.IMPRECISE\_FILTERING**

Number of instruction blocks traced on a transition in the filtering.

Type: `uint8_t`

Default value: 0

#### **`ete.LPOVERRIDE`**

Low power override.

Type: `bool`

Default value: `true`

#### **`ete.MAXSPEC`**

Maximum speculation depth.

Type: `uint32_t`

Default value: 0x0

#### **`ete.MAX_INST_PER_Q`**

Maximum limit for the number of instructions implied by a Q element.

Type: `uint16_t`

Default value: 0x1

#### **`ete.NOOVERFLOW`**

No overflow.

Type: `bool`

Default value: `false`

#### **`ete.NUMACPAIRS`**

Number of instruction address comparators pairs.

Type: `uint8_t`

Default value: 4

#### **`ete.NUMCIDC`**

Number of context ID comparators.

Type: `uint8_t`

Default value: 1

#### **`ete.NUMCNTR`**

Number of counters.

Type: `uint8_t`

Default value: 2

**`ete.NUMEXTINSEL`**

Number of external input selectors.

Type: `uint8_t`

Default value: 4

**`ete.NUMPC`**

Number of PE comparators.

Type: `uint8_t`

Default value: 0

**`ete.NUMSEQSTATE`**

Number of sequencer states.

Type: `uint8_t`

Default value: 4

**`ete.NUMSSCC`**

Number of single shot comparators.

Type: `uint8_t`

Default value: 1

**`ete.NUMVMIDC`**

Number of virtual ID comparators.

Type: `uint8_t`

Default value: 1

**`ete.NumberOfETEEEvents`**

Number of trace events.

Type: `uint8_t`

Default value: 2

**`ete.NumberOfRSPairs`**

Number of resource selector pairs.

Type: `uint8_t`

Default value: 8

#### **`ete.PIDR_CM0D`**

TRCPIDR CM0D value.

Type: `uint8_t`

Default value: 0

#### **`ete.PIDR_DESIGNER`**

TRCPIDR DESIGNER value.

Type: `uint16_t`

Default value: 0x0

#### **`ete.PIDR_PART`**

TRCPIDR PART number value.

Type: `uint16_t`

Default value: 0x0

#### **`ete.PIDR_REVAND`**

TRCPIDR REVAND value.

Type: `uint8_t`

Default value: 0

#### **`ete.PIDR_REVISION`**

TRCPIDR REVISION value.

Type: `uint8_t`

Default value: 0

#### **`ete.QFILT`**

Q filtering.

Type: `bool`

Default value: `false`

#### **`ete.QSUP`**

Q support.



Type: `uint8_t`

Default value: 0

### **`ete.Q_CADENCE`**

Number of instruction blocks traced between two Q elements.

Type: `uint16_t`

Default value: 0x1

### **`ete.RAZWI_REG_SEL_TOP_BIT`**

Implement Resource Selectors or Resource Selector Pairs bits as **RAZ/WI**.

Type: `bool`

Default value: `false`

### **`ete.REGS_WRITE_IGNORE_WHEN_ENABLED`**

Whether direct and external writes to registers except TRCPRGCTLR, TRCCLAIMSET and TRCCLAIMCLR are ignored when not in Idle state.

Type: `bool`

Default value: `false`

### **`ete.REG_ACCESS_ONLY_MODE`**

If enabled, all traces are disabled. Plugin only allows register accesses.

Type: `bool`

Default value: `false`

### **`ete.RES0_STATEFUL`**

Whether **RES0** bits are stateful or **RAZ/WI**.

Type: `bool`

Default value: `false`

### **`ete.RETSTACK`**

Return stack depth.

Type: `uint8_t`

Default value: 3

**ete.REVISION**

TRCIDR1 revision value.

Type: `uint8_t`

Default value: 0

**ete.SIM\_OVERFLOW\_GRANULARITY**

Number of instruction blocks in each granule, for simulated overflow.

Type: `uint32_t`

Default value: `0x64`

**ete.SIM\_OVERFLOW\_PERCENTAGE**

Percentage of instruction blocks lost in each granule, for simulated overflow.

Type: `uint8_t`

Default value: 0

**ete.SOURCE\_ADDRESS**

Allow generation of source address elements.

Type: `bool`

Default value: `false`

**ete.STALLCTRL**

Stall control.

Type: `bool`

Default value: `true`

**ete.SYSTALL**

System stall.

Type: `bool`

Default value: `true`

**ete.TRACEIDSIZE**

Trace ID size.

Type: `uint8_t`

Default value: 7

**ete.TRACE\_OUTPUT**

File to which to write trace byte stream.

Type: `string`

Default value: `N/A`

**ete.TRACE\_OUTPUT\_ENABLE**

ETE Trace output enable: 1=enable, 0=disable.

Type: `bool`

Default value: `false`

**ete.TRCSRSTA\_FORCED\_EXCEP**

TRCSRSTA value for a forcibly traced exception.

Type: `bool`

Default value: `false`

**ete.TSMARK**

Whether timestamp markers are supported.

Type: `bool`

Default value: `false`

**ete.TSSIZE**

Timestamp size.

Type: `uint8_t`

Default value: `8`

**ete.WFXMODE**

WFX mode.

Type: `bool`

Default value: `true`

**ets\_level**

Level of Enhanced Translation Synchronization support. 0: FEAT\_ETS2 not supported, 1: FEAT\_ETS2 not supported, 2: FEAT\_ETS2 supported, 3: FEAT\_ETS3 supported.

Type: `uint8_t`

Default value: 0

#### **exception\_catch\_before\_software\_step**

Exception catch priority for the exception trapping form of exception catch (Armv8.2 or later, or exception\_catch\_type=0). If true, the exception catch debug event has higher priority than software step and halting step.

Type: bool

Default value: true

#### **exception\_catch\_type**

Type of exception catch (Armv8.0 - Armv8.1 only). 0, exception trapping. 1, non-exception trapping, higher priority than step. 2, non-exception trapping, lower priority than step.

Type: uint8\_t

Default value: 0

#### **exclusive\_monitor\_clear\_on\_atomic\_from\_same\_master**

Exclusive monitors in the cluster will be cleared by a atomic by the same master to the monitored address.

Type: bool

Default value: true

#### **exclusive\_monitor\_clear\_on\_store\_from\_same\_master**

Exclusive monitors in the cluster will be cleared by a store by the same master to the monitored address.

Type: bool

Default value: true

#### **exclusive\_monitor\_clear\_on\_strex\_address\_mismatch**

Exclusive monitors in the cluster will be cleared when a strex fails because the address does not match.

Type: bool

Default value: true

#### **exclusive\_monitor\_clear\_on\_strex\_success**

Exclusive monitors in the cluster will be cleared when a strex succeeds.

Type: bool

Default value: `true`

### **`exercise_stxr_fail`**

Controls the rejection of exclusive store instructions. 0: exclusive store instructions should behave as normal, 1: Reject a pseudo-random majority of exclusive store instructions, 2: Always fail exclusive store instructions.

Type: `uint8_t`

Default value: 0

### **`ext_abort_device_GRE_prefetch_ras_index`**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as `ext_abort_prefetch_ras_index`, Valid indices in range [0, `number_of_error_records`-1].

Type: `int16_t`

Default value: -1

### **`ext_abort_device_GRE_prefetch_ras_type`**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as `ext_abort_prefetch_ras_type`, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `int8_t`

Default value: -1

### **`ext_abort_device_GRE_read_is_critical`**

Critical reporting of device-GRE read external aborts.

Type: `bool`

Default value: `false`

### **`ext_abort_device_GRE_read_is_sync`**

Synchronous reporting of device-GRE read external aborts. 0, asynchronous. 1, synchronous. 2, same as `ext_abort_device_read_is_sync`.

Type: `uint8_t`

Default value: 2

### **`ext_abort_device_GRE_read_ras_index`**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as `ext_abort_device_read_ras_index`, Valid indices in range [0, `number_of_error_records`-1].

Type: `int16_t`

Default value: -1

#### **ext\_abort\_device\_GRE\_read\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as ext\_abort\_device\_read\_ras\_type, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `int8_t`

Default value: -1

#### **ext\_abort\_device\_GRE\_write\_is\_critical**

Critical reporting of device-GRE write external aborts.

Type: `bool`

Default value: `false`

#### **ext\_abort\_device\_GRE\_write\_is\_sync**

Synchronous reporting of device-GRE write external aborts. 0, asynchronous. 1, synchronous. 2, same as ext\_abort\_device\_write\_is\_sync.

Type: `uint8_t`

Default value: 2

#### **ext\_abort\_device\_GRE\_write\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as ext\_abort\_device\_write\_ras\_index, Valid indices in range [0, number\_of\_error\_records-1].

Type: `int16_t`

Default value: -1

#### **ext\_abort\_device\_GRE\_write\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as ext\_abort\_device\_write\_ras\_type, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `int8_t`

Default value: -1

#### **ext\_abort\_device\_nGRE\_prefetch\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as ext\_abort\_prefetch\_ras\_index, Valid indices in range [0, number\_of\_error\_records-1].

Type: `int16_t`

Default value: -1

**ext\_abort\_device\_nGRE\_prefetch\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as ext\_abort\_prefetch\_ras\_type, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `int8_t`

Default value: -1

**ext\_abort\_device\_nGRE\_read\_is\_critical**

Critical reporting of device-nGRE read external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_device\_nGRE\_read\_is\_sync**

Synchronous reporting of device-nGRE read external aborts. 0, asynchronous. 1, synchronous. 2, same as ext\_abort\_device\_read\_is\_sync.

Type: `uint8_t`

Default value: 2

**ext\_abort\_device\_nGRE\_read\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as ext\_abort\_device\_read\_ras\_index, Valid indices in range [0, number\_of\_error\_records-1].

Type: `int16_t`

Default value: -1

**ext\_abort\_device\_nGRE\_read\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as ext\_abort\_device\_read\_ras\_type, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `int8_t`

Default value: -1

**ext\_abort\_device\_nGRE\_write\_is\_critical**

Critical reporting of device-nGRE write external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_device\_nGRE\_write\_is\_sync**

Synchronous reporting of device-nGRE write external aborts. 0, asynchronous. 1, synchronous. 2, same as ext\_abort\_device\_write\_is\_sync.

Type: `uint8_t`

Default value: 2

**ext\_abort\_device\_nGRE\_write\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as ext\_abort\_device\_write\_ras\_index, Valid indices in range [0, number\_of\_error\_records-1].

Type: `int16_t`

Default value: -1

**ext\_abort\_device\_nGRE\_write\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as ext\_abort\_device\_write\_ras\_type, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `int8_t`

Default value: -1

**ext\_abort\_device\_prefetch\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as ext\_abort\_prefetch\_ras\_index, Valid indices in range [0, number\_of\_error\_records-1].

Type: `int16_t`

Default value: -1

**ext\_abort\_device\_prefetch\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as ext\_abort\_prefetch\_ras\_type, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `int8_t`

Default value: -1

**ext\_abort\_device\_read\_acquire\_is\_sync**

Synchronous reporting of device read with acquire external aborts.

Type: `bool`

Default value: `false`



**ext\_abort\_device\_read\_is\_critical**

Critical reporting of device-nGnRE read external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_device\_read\_is\_sync**

Synchronous reporting of device-nGnRE read external aborts.

Type: `bool`

Default value: `true`

**ext\_abort\_device\_read\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number\_of\_error\_records-1].

Type: `uint16_t`

Default value: `0x0`

**ext\_abort\_device\_read\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `uint8_t`

Default value: `0`

**ext\_abort\_device\_write\_is\_critical**

Critical reporting of device-nGnRE write external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_device\_write\_is\_sync**

Synchronous reporting of device-nGnRE write external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_device\_write\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number\_of\_error\_records-1].

Type: `uint16_t`

Default value: `0x0`

### **ext\_abort\_device\_write\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `uint8_t`

Default value: `0`

### **ext\_abort\_fill\_data**

Returned data, if external aborts are asynchronous.

Type: `uint64_t`

Default value: `0xfdfdfdfdfcfdfdfd`

### **ext\_abort\_in\_virtual\_tag\_reads\_is\_sync**

Behaviour of external aborts generated by virtual tag reads. 0, asynchronous abort. 1, synchronous abort. 2, same as `ext_abort_normal_cacheable_read_is_sync`.

Type: `uint8_t`

Default value: `2`

### **ext\_abort\_in\_virtual\_tag\_writes\_is\_sync**

Behaviour of external aborts generated by virtual tag writes. 0, asynchronous abort. 1, synchronous abort. 2, same as `ext_abort_normal_cacheable_write_is_sync`.

Type: `uint8_t`

Default value: `2`

### **ext\_abort\_normal\_cacheable\_read\_is\_critical**

Critical reporting of normal write-back cacheable-read external aborts.

Type: `bool`

Default value: `false`

### **ext\_abort\_normal\_cacheable\_read\_is\_sync**

Synchronous reporting of normal write-back cacheable-read external aborts.

Type: `bool`

Default value: `true`

**ext\_abort\_normal\_cacheable\_read\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number\_of\_error\_records-1].

Type: `uint16_t`

Default value: `0x0`

**ext\_abort\_normal\_cacheable\_read\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `uint8_t`

Default value: `0`

**ext\_abort\_normal\_cacheable\_write\_is\_critical**

Critical reporting of normal write-back cacheable write external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_normal\_cacheable\_write\_is\_sync**

Synchronous reporting of normal write-back cacheable write external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_normal\_cacheable\_write\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number\_of\_error\_records-1].

Type: `uint16_t`

Default value: `0x0`

**ext\_abort\_normal\_cacheable\_write\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `uint8_t`

Default value: `0`

**ext\_abort\_normal\_noncacheable\_prefetch\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as ext\_abort\_prefetch\_ras\_index, Valid indices in range [0, number\_of\_error\_records-1].

Type: int16\_t

Default value: -1

**ext\_abort\_normal\_noncacheable\_prefetch\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as ext\_abort\_prefetch\_ras\_type, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: int8\_t

Default value: -1

**ext\_abort\_normal\_noncacheable\_read\_is\_critical**

Critical reporting of normal noncacheable-read external aborts.

Type: bool

Default value: false

**ext\_abort\_normal\_noncacheable\_read\_is\_sync**

Synchronous reporting of normal noncacheable-read external aborts.

Type: bool

Default value: true

**ext\_abort\_normal\_noncacheable\_read\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number\_of\_error\_records-1].

Type: uint16\_t

Default value: 0x0

**ext\_abort\_normal\_noncacheable\_read\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: uint8\_t

Default value: 0

**ext\_abort\_normal\_noncacheable\_write\_is\_critical**

Critical reporting of normal noncacheable write external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_normal\_noncacheable\_write\_is\_sync**

Synchronous reporting of normal noncacheable write external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_normal\_noncacheable\_write\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number\_of\_error\_records-1].

Type: `uint16_t`

Default value: `0x0`

**ext\_abort\_normal\_noncacheable\_write\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `uint8_t`

Default value: `0`

**ext\_abort\_normal\_wt\_cacheable\_prefetch\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as `ext_abort_prefetch_ras_index`, Valid indices in range [0, number\_of\_error\_records-1].

Type: `int16_t`

Default value: `-1`

**ext\_abort\_normal\_wt\_cacheable\_prefetch\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as `ext_abort_prefetch_ras_type`, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `int8_t`

Default value: `-1`

**ext\_abort\_normal\_wt\_cacheable\_read\_is\_critical**

Critical reporting of normal write-through cacheable-read external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_normal\_wt\_cacheable\_read\_is\_sync**

Synchronous reporting of normal write-through read external aborts. 0, asynchronous. 1, synchronous. 2, same as `ext_abort_normal_cacheable_read_is_sync`.

Type: `uint8_t`

Default value: 2

**ext\_abort\_normal\_wt\_cacheable\_read\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as `ext_abort_normal_cacheable_read_ras_index`, Valid indices in range [0, `number_of_error_records`-1].

Type: `int16_t`

Default value: -1

**ext\_abort\_normal\_wt\_cacheable\_read\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as `ext_abort_normal_cacheable_read_ras_type`, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `int8_t`

Default value: -1

**ext\_abort\_normal\_wt\_cacheable\_write\_is\_critical**

Critical reporting of normal write-through write external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_normal\_wt\_cacheable\_write\_is\_sync**

Synchronous reporting of normal write-through write external aborts. 0, asynchronous. 1, synchronous. 2, same as `ext_abort_normal_cacheable_write_is_sync`.

Type: `uint8_t`

Default value: 2

**ext\_abort\_normal\_wt\_cacheable\_write\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as ext\_abort\_normal\_cacheable\_write\_ras\_index, Valid indices in range [0, number\_of\_error\_records-1].

Type: int16\_t

Default value: -1

**ext\_abort\_normal\_wt\_cacheable\_write\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as ext\_abort\_normal\_cacheable\_write\_ras\_type, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: int8\_t

Default value: -1

**ext\_abort\_prefetch\_device\_GRE\_read\_is\_critical**

Critical reporting of external aborts generated by device-GRE instruction fetches.

Type: bool

Default value: false

**ext\_abort\_prefetch\_device\_GRE\_read\_is\_sync**

Behaviour of external aborts generated by device-GRE instruction fetches. 0, asynchronous abort. 1, synchronous abort. 2, same as ext\_abort\_prefetch\_is\_sync.

Type: uint8\_t

Default value: 2

**ext\_abort\_prefetch\_device\_nGRE\_read\_is\_critical**

Critical reporting of external aborts generated by device-nGRE instruction fetches.

Type: bool

Default value: false

**ext\_abort\_prefetch\_device\_nGRE\_read\_is\_sync**

Behaviour of external aborts generated by device-nGRE instruction fetches. 0, asynchronous abort. 1, synchronous abort. 2, same as ext\_abort\_prefetch\_is\_sync.

Type: uint8\_t

Default value: 2

**ext\_abort\_prefetch\_device\_read\_is\_critical**

Critical reporting of external aborts generated by device-nGnRE instruction fetches.

Type: `bool`

Default value: `false`

**ext\_abort\_prefetch\_device\_read\_is\_sync**

Behaviour of external aborts generated by device-nGnRE instruction fetches. 0, asynchronous abort. 1, synchronous abort. 2, same as `ext_abort_prefetch_is_sync`.

Type: `uint8_t`

Default value: 2

**ext\_abort\_prefetch\_is\_critical**

Critical reporting of external aborts generated by normal writeback cacheable instruction fetches.

Type: `bool`

Default value: `false`

**ext\_abort\_prefetch\_is\_sync**

Behaviour of external aborts generated by normal writeback cacheable instruction fetches. 0, asynchronous abort. 1, synchronous abort.

Type: `bool`

Default value: `true`

**ext\_abort\_prefetch\_noncacheable\_read\_is\_critical**

Critical reporting of external aborts generated by normal noncacheable instruction fetches.

Type: `bool`

Default value: `false`

**ext\_abort\_prefetch\_noncacheable\_read\_is\_sync**

Behaviour of external aborts generated by normal noncacheable instruction fetches. 0, asynchronous abort. 1, synchronous abort. 2, same as `ext_abort_prefetch_is_sync`.

Type: `uint8_t`

Default value: 2



**ext\_abort\_prefetch\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number\_of\_error\_records-1].

Type: `uint16_t`

Default value: `0x0`

**ext\_abort\_prefetch\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `uint8_t`

Default value: `0`

**ext\_abort\_prefetch\_so\_read\_is\_critical**

Critical reporting of external aborts generated by device-nGnRnE instruction fetches.

Type: `bool`

Default value: `false`

**ext\_abort\_prefetch\_so\_read\_is\_sync**

Behaviour of external aborts generated by device=nGnRnE instruction fetches. 0, asynchronous abort. 1, synchronous abort. 2, same as `ext_abort_prefetch_is_sync`.

Type: `uint8_t`

Default value: `2`

**ext\_abort\_prefetch\_wt\_cacheable\_read\_is\_critical**

Critical reporting of external aborts generated by normal writethrough cacheable instruction fetches.

Type: `bool`

Default value: `false`

**ext\_abort\_prefetch\_wt\_cacheable\_read\_is\_sync**

Behaviour of external aborts generated by normal writethrough cacheable instruction fetches. 0, asynchronous abort. 1, synchronous abort. 2, same as `ext_abort_prefetch_is_sync`.

Type: `uint8_t`

Default value: `2`

**ext\_abort\_so\_prefetch\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as ext\_abort\_prefetch\_ras\_index, Valid indices in range [0, number\_of\_error\_records-1].

Type: `int16_t`

Default value: -1

**ext\_abort\_so\_prefetch\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as ext\_abort\_prefetch\_ras\_type, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `int8_t`

Default value: -1

**ext\_abort\_so\_read\_is\_critical**

Critical reporting of device-nGnRnE read external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_so\_read\_is\_sync**

Synchronous reporting of device-nGnRnE read external aborts.

Type: `bool`

Default value: `true`

**ext\_abort\_so\_read\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number\_of\_error\_records-1].

Type: `uint16_t`

Default value: 0x0

**ext\_abort\_so\_read\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `uint8_t`

Default value: 0

**ext\_abort\_so\_write\_is\_critical**

Critical reporting of device-nGnRnE write external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_so\_write\_is\_sync**

Synchronous reporting of device-nGnRnE write external aborts.

Type: `bool`

Default value: `true`

**ext\_abort\_so\_write\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number\_of\_error\_records-1].

Type: `uint16_t`

Default value: `0x0`

**ext\_abort\_so\_write\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `uint8_t`

Default value: `0`

**ext\_abort\_ttw\_cacheable\_read\_is\_critical**

Critical reporting of TTW cacheable read external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_ttw\_cacheable\_read\_is\_sync**

Synchronous reporting of TTW cacheable read external aborts.

Type: `bool`

Default value: `true`

**ext\_abort\_ttw\_cacheable\_read\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number\_of\_error\_records-1].

Type: `uint16_t`

Default value: `0x0`

#### **`ext_abort_ttw_cacheable_read_ras_type`**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `uint8_t`

Default value: `0`

#### **`ext_abort_ttw_noncacheable_read_is_critical`**

Critical reporting of TTW noncacheable read external aborts.

Type: `bool`

Default value: `false`

#### **`ext_abort_ttw_noncacheable_read_is_sync`**

Synchronous reporting of TTW noncacheable read external aborts.

Type: `bool`

Default value: `true`

#### **`ext_abort_ttw_noncacheable_read_ras_index`**

External Aborts are reported in RAS record index specified in this param. Values: Valid indices in range [0, number\_of\_error\_records-1].

Type: `uint16_t`

Default value: `0x0`

#### **`ext_abort_ttw_noncacheable_read_ras_type`**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `uint8_t`

Default value: `0`

#### **`ext_abort_ttw_wt_cacheable_read_is_critical`**

Critical reporting of TTW write-through cacheable read external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_ttw\_wt\_cacheable\_read\_is\_sync**

Synchronous reporting of TTW write-through cacheable read external aborts. 0, asynchronous. 1, synchronous. 2, same as ext\_abort\_ttw\_cacheable\_read\_is\_sync.

Type: `uint8_t`

Default value: 2

**ext\_abort\_ttw\_wt\_cacheable\_read\_ras\_index**

External Aborts are reported in RAS record index specified in this param. Values: -1 = Same as ext\_abort\_ttw\_cacheable\_read\_ras\_index, Valid indices in range [0, number\_of\_error\_records-1].

Type: `int16_t`

Default value: -1

**ext\_abort\_ttw\_wt\_cacheable\_read\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: -1 = Same as ext\_abort\_ttw\_cacheable\_read\_ras\_type, 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `int8_t`

Default value: -1

**external\_debug\_request\_delay**

Configure External Debug Request delay in CPU cycles.

Type: `uint32_t`

Default value: 0x0

**external\_oslar\_access\_disabled\_by\_authentication**

If true, external accesses to OSLAR, when external debugging is not enabled, will generate an error (FEAT\_Debugv8p2).

Type: `bool`

Default value: `false`

**far\_unchanged\_for\_stack\_alignment\_fault**

If true and a Stack Pointer Alignment Fault occurs, FAR\_ELx is not updated, whatever value it had is maintained.

Type: `bool`

Default value: `false`

**far\_unchanged\_for\_virtual\_serror**

If true and a virtual SError occurs, FAR\_ELx is not updated, whatever value it had is maintained.

Type: `bool`

Default value: `false`

**far\_unchanged\_when\_fnv\_set**

If true and ESR\_ELx.FnV=1, FAR\_ELx is not updated but PFAR\_ELx/MFAR\_ELx can be updated.

Type: `bool`

Default value: `false`

**fault\_on\_misprogrammed\_gpt\_contig\_region**

Whether GPF faults occur when GPT contiguous entries are misprogrammed.

Type: `bool`

Default value: `false`

**fault\_on\_nT\_bit\_set**

Whether block translation table entries with the nT bit set should always fault. Only applies when `changing_block_size_without_bbm_support_level` is 1 or higher.

Type: `bool`

Default value: `true`

**fault\_unalign\_to\_unsupported\_access**

If `has_unaligned_single_copy_atomicity` is true, whether unaligned A64 atomic, exclusive and acquire/release instructions to non iWB-oWB or the access crossing a 16-byte boundary generate fault. Bits 0,1,2,3 should be set accordingly to enable the fault behaviour. bit 0: atomic access should fault, bit 1: exclusive access should fault, bit 2: acquire/release should fault, bit 3: the access crossing a 16-byte boundary should fault.

Type: `uint8_t`

Default value: 8

**fault\_unaligned\_s1\_device\_s2\_fwb**

Whether unaligned fault with stage1 Device memory and final memory attribute forced to normal by FWB. 0, No fault. 1, will fault. 2 No fault if final Shareability is NSH.

Type: `uint8_t`

Default value: 0

**fgdt\_index\_width**

Implemented FGDT Index width value, 0: 3 bits. 1: 4 bits. 2: 5 bits.

Type: `uint8_t`

Default value: 0

**force\_align\_pc**

**UNPREDICTABLE** branch to non-word-aligned address in ARM state is forced to be aligned.

Type: `bool`

Default value: `false`

**force\_deterministic\_irg\_tag\_generation**

Force the random tag generated by the IRG instruction when GCR\_EL1.RRND=1 to equal RGSRR\_EL1.SEED[3:0] rather than a non-deterministic value.

Type: `bool`

Default value: `false`

**force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks**

Force MTE tag accesses to **RAZ/WI** and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

Type: `bool`

Default value: `false`

**force\_pstate\_pm\_in\_debug\_state**

If true, PSTATE.PM is forced to 1 while entering in debug state (FEAT\_EBEP).

Type: `bool`

Default value: `false`

**force\_sync\_on\_wfx**

If true, the PE does a context synchronization before entering low power state(WFI/WFE).

Type: `bool`

Default value: `false`

**force\_wnr\_read\_unsupported\_exclusive\_or\_atomic**

DEPRECATED, please use `force_wnr_read_unsupported_exclusive_or_atomic` instead. Whether ESR\_ELx.WnR is forced to 0 for unsupported atomic and exclusives.

Type: `bool`

Default value: `false`

**force\_wnr\_unsupported\_atomic\_hwu**

If not 0, force ESR\_ELx.WnR to a specific value on Unsupported Atomic Hardware Update. Possible values: 0: Not forced. 1: Write. 2: Read.

Type: `uint8_t`

Default value: 0

**force\_wnr\_unsupported\_exclusive\_or\_atomic**

If not 0, force ESR\_ELx.WnR to a specific value on Unsupported Atomic or Exclusives. Possible values: 0: Not forced. 1: Write. 2: Read.

Type: `uint8_t`

Default value: 0

**force\_zero\_PSTATE\_PAN**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

**force\_zero\_mpam\_partid\_and\_pmg**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

Type: `uint8_t`

Default value: 0

**fp8\_support\_level**

0->No support for Advanced SIMD, SVE2 FP8 instructions 1->Support for FEAT\_FP8 2->Support for FEAT\_FP8FMA 3->Support for FEAT\_FP8DOT4 4->Support for FEAT\_FP8DOT2.

Type: `uint8_t`

Default value: 0



**fpcr\_short\_vector\_raz**

FPSCR and FPCR fields LEN and STRIDE are hardwired to 0.

Type: `bool`

Default value: `false`

**fpsr\_res0\_stateful\_mask**

Mask for stateful bits of FPSR which are **RES0**.

Type: `uint32_t`

Default value: `0x0`

**fsr\_ext\_bit\_update\_kind**

Set/Clear DFSR/IFSR EA bit on Synchronous/Async External Aborts. 0: Never Set, 1: Set on Synchronous Ext Aborts 2: Set on Asynchronous Ext Aborts 3: Set on both Sync and Async Ext Aborts.

Type: `uint8_t`

Default value: `3`

**gcs\_data\_check\_overrides\_data\_abort**

GCS Data check exceptions are taken before Data Aborts.

Type: `bool`

Default value: `false`

**gcs\_overshoot\_writes**

Number of overshooting GCS records written after a writing a record.

Type: `uint64_t`

Default value: `0x0`

**gcspr\_sync\_immediate**

If true, writing to GCSPR\_ELx registers has immediate effect regardless of `has_delayed_sysreg` flag.

Type: `bool`

Default value: `false`

**gic.GICC-offset**

Offset from PERIPHBASE for GICC registers.

Type: `uint32_t`

Default value: `0x2000`

#### **`gic.GICD-offset`**

Offset from PERIPHBASE for GICD registers. Will be ignored when GICv3 CPU interface is enabled, as distributor is then external to the cluster.

Type: `uint32_t`

Default value: `0x1000`

#### **`gic.GICH-offset`**

Offset from PERIPHBASE for GICH registers.

Type: `uint32_t`

Default value: `0x4000`

#### **`gic.GICH-other-CPU-offset`**

Offset from PERIPHBASE for GICH registers for accessing other CPUs in the cluster. Set to 0 to disable.

Type: `uint32_t`

Default value: `0x5000`

#### **`gic.GICV-alias`**

Offset from PERIPHBASE for alias of GICV registers. When `gicv2-only`, if zero no alias will be created; if `gicv2-only=0`, the param is deprecated, when zero or unset an alias is created in the place mandated by the architecture (`GICV-base+0xF000`).

Type: `uint32_t`

Default value: `0x0`

#### **`gic.GICV-offset`**

Offset from PERIPHBASE for GICV registers.

Type: `uint32_t`

Default value: `0x6000`

#### **`gic.PERIPH-size`**

Size of registers based at PERIPHBASE that are considered to be owned by the GIC. Any accesses in the range PERIPHBASE to PERIPHBASE+`gic.PERIPH-size-1` that do not match GIC registers will be treated as **RAZ/WI**.

Type: `uint32_t`

Default value: `0x8000`

### **`gicv3.A3-affinity-supported`**

Whether a non-zero value for affinity at level 3 is supported.

Type: `bool`

Default value: `false`

### **`gicv3.BPR-min`**

The minimum value for the GICC\_BPR register (non-secure version will be 1 + this value).

Type: `uint8_t`

Default value: `2`

### **`gicv3.EOI-check-CPUID`**

Check CPU ID specified for accesses to EOI registers (rather than just ending highest priority active interrupt).

Type: `bool`

Default value: `false`

### **`gicv3.EOI-check-ID`**

Check Interrupt ID specified for accesses to EOI registers (rather than just ending highest priority active interrupt).

Type: `bool`

Default value: `false`

### **`gicv3.EOI-deactivate-any-interrupt`**

Allow an EOI to deactivate interrupts that aren't the highest priority active interrupt (EOI-ignore-out-of-order must be false otherwise this is ignored).

Type: `bool`

Default value: `false`

### **`gicv3.EOI-ignore-out-of-order`**

Ignore EOI writes that cannot end the highest priority active interrupt.

Type: `bool`

Default value: `true`

**`gicv3.FIQEn-RAO`**

GICC\_CTLR.FIQEn is read as one, write insensitive.

Type: `bool`

Default value: `false`

**`gicv3.IIDR_base`**

The base value for calculating the GICC\_IIDR register value.

Type: `uint32_t`

Default value: `0x43b`

**`gicv3.LR-count`**

The number of implemented list registers.

Type: `uint8_t`

Default value: `16`

**`gicv3.PMHE-RAO-WI`**

ICC\_CTLR\_EL\*.PHME is read as one, write insensitive.

Type: `bool`

Default value: `false`

**`gicv3.PMHE-RAZ-WI`**

ICC\_CTLR\_EL\*.PHME is read as zero, write insensitive.

Type: `bool`

Default value: `false`

**`gicv3.PMHE-release-set-packet`**

if PHME is enabled, whether a SET packet is released by CPU Intf in Upstream Ack window.

Type: `bool`

Default value: `false`

**`gicv3.SRE-EL2-enable-RAO`**

When ICC\_SRE\_EL2.SRE is **RAO/WI**, makes ICC\_SRE\_EL2.Enable **RAO/WI**.

Type: `bool`

Default value: `false`

**gicv3.SRE-EL3-enable-RAO**

When ICC\_SRE\_EL3.SRE is **RAO/WI**, makes ICC\_SRE\_EL3.Enable **RAO/WI**.

Type: `bool`

Default value: `false`

**gicv3.SRE-EL3-set-once**

Restrict SRE EL3 to be set only once.

Type: `bool`

Default value: `false`

**gicv3.SRE-enable-action-on-mmap**

Allowed values are: 0-SRE one allows mmap access. 1-SRE one disables mmap access. 2-SRE one makes mmap access **RAZ-WI**.

Type: `uint8_t`

Default value: `0`

**gicv3.STATUSR-implemented**

If GICv3 CPU interface is being used, this determines whether the STATUS registers are implemented.

Type: `bool`

Default value: `true`

**gicv3.VBPR-min**

The minimum value for the GICV\_BPR register (non-secure version will be 1 + this value).

Type: `uint8_t`

Default value: `2`

**gicv3.VFIQEn-RAO**

ICH\_VMCR\_EL2.VFIQEn is read as one, write insensitive.

Type: `bool`

Default value: `false`

**gicv3.cputnf-mmap-access-level**

Allowed values are: 0-mmap access is supported for GICC,GICH,GICV registers. 1-mmap access is supported only for GICV registers. 2-mmap access is not supported.

Type: `uint8_t`

Default value: `0`

### **`gicv3.dir-trap-support`**

The cpu supports separate trapping of ICC\_DIR\_EL1 to EL2.

Type: `bool`

Default value: `true`

### **`gicv3.el3_trap_priority_when_secure_debug_disabled`**

Undef to access priorities group register when secure debug is disabled.

Type: `bool`

Default value: `false`

### **`gicv3.extended-interrupt-range-support`**

Device has support for extended SPI/PPI ID ranges.

Type: `bool`

Default value: `false`

### **`gicv3.gicv2-only`**

Limit the GIC implementation to GICv2 features only.

Type: `bool`

Default value: `false`

### **`gicv3.idle-is-ff`**

For GICC/GICV RPR, when idle, return FF when true, minimum supported priority otherwise.

Type: `bool`

Default value: `true`

### **`gicv3.ignore-DIR-write-when-EOImode-not-set`**

Ignore **UNPREDICTABLE** access to GICC\_DIR register.

Type: `bool`

Default value: `true`

### **`gicv3.interrupt-bypass-support`**

Interrupt bypass support, set to false for devices not supporting interrupt bypass.

Type: `bool`

Default value: `true`

### **`gicv3.local-SEIs`**

Generate SEI to signal internal issues.

Type: `bool`

Default value: `false`

### **`gicv3.local-VSEIs`**

Generate VSEI to signal internal issues.

Type: `bool`

Default value: `false`

### **`gicv3.physical-ID-bits`**

Number of physical ID bits implemented.

Type: `uint8_t`

Default value: 16

### **`gicv3.priority-bits`**

Number of priority bits implemented.

Type: `uint8_t`

Default value: 5

### **`gicv3.send-PMHE-command-only-when-priority-changes`**

Send PMHE upstream command to distributor only when write to ICC\_PMR\_EL1 changes the priority.

Type: `bool`

Default value: `false`

### **`gicv3.sgi-range-selector-support`**

Device has support for the Range Selector feature for SGI.

Type: `bool`

Default value: `false`

**gicv3.suppress-virtual-enables-comms**

In GICv3 only mode, prevents the GIC CPUIF from communicating UpstreamWrite/VirtualEnables to the IRI.

Type: `bool`

Default value: `true`

**gicv3.virtual-ID-bits**

Number of virtual ID bits implemented.

Type: `uint8_t`

Default value: `16`

**gicv3.virtual-lpi-support**

When GICv3 is supported, indicates a cut down CPUIF interface with no support of VLPI (GICv3 only) when false.

Type: `bool`

Default value: `true`

**gicv3.virtual-priority-bits**

Number of virtual priority bits implemented.

Type: `uint8_t`

Default value: `5`

**gicv3.without-DS-support**

GICv3 CPU interfaces do not support disabling security in the distributor (GICD\_CTLR.DS=1).

Type: `bool`

Default value: `false`

**gicv4.mask-virtual-interrupt**

If true, virtual interrupts can be masked from being reported to virtual CPU interface by setting ICH\_HCR\_EL2.DVIM 1. No control otherwise.

Type: `bool`

Default value: `false`

**gicv5.config\_file**

File path for the GICv5 configuration yaml. The file lists the GICv5 params.



Type: `string`

Default value: `N/A`

### **`gicv5.has_gcie_legacy`**

When set to true, FEAT\_GCIE\_LEGACY is supported.

Type: `bool`

Default value: `false`

### **`gicv5.interrupt-bypass-support`**

Interrupt bypass support. when set to true, bypasses GICv5 CPU interface to signal interrupts to the PE.

Type: `bool`

Default value: `false`

### **`global_debug_rom.ROMDEVID`**

Value of Debug Rom Device Identification Register.

Type: `uint64_t`

Default value: `0x0`

### **`global_debug_rom.ROMPIDR`**

Value of Debug Rom Peripheral Identification Register.

Type: `uint64_t`

Default value: `0x4000bb000`

### **`global_debug_rom.ROMPRIDR0`**

Value of Debug ROM Power RequestID Register.

Type: `uint8_t`

Default value: `1`

### **`gpccr_el3_gpcp_behaviour`**

Used to control impdef behaviour when GPCP=1 (0->Faults are always generated and reported, 1->Faults are not generated and reported), 2->Faults are generated and reported only for Arm recommended cases.

Type: `uint8_t`

Default value: `2`

**gpt\_tlb\_size**

Number of separate GPT TLB entries.

Type: `uint32_t`

Default value: `0x0`

**gpt\_walkcache\_size**

Number of GPT walk cache entries.

Type: `uint32_t`

Default value: `0x0`

**hardware\_translation\_table\_update\_implemented**

Implement hardware translation table updates from ARMv8.1 (FEAT\_HAFDBS).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.1 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: `1`

**has-gicv4.1**

GICv4.1 is enabled, and all the features with GICv4.1 are implemented (FEAT\_GICv4p1).

Type: `bool`

Default value: `false`

**has\_128\_bit\_atomic\_instructions**

Implement 128-bit Atomic Instructions (FEAT\_LSE128); mandatory in the presence of Future Architecture Technologies (FAT)values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: `0`

**has\_128\_bit\_tt\_descriptors**

Implement 128-bit Translation Table Descriptors (FEAT\_D128)values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: `0`

**has\_16bit\_asids**

Enable 16-bit ASIDs; mandatory in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**has\_16bit\_vmids**

Implement support for 16-bit VMIDs from ARMv8.1 (FEAT\_VMID16).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.1 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: `1`

**has\_16k\_granule**

Implement the 16k LPAE translation granule.

Type: `bool`

Default value: `false`

**has\_4k\_granule**

Implement the 4k LPAE translation granule.

Type: `bool`

Default value: `true`

**has\_52bit\_address\_with\_16k**

Implements Armv8.7 52-bit IPA/PA support for 16k (FEAT\_LPA2).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.7 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: `0`

**has\_52bit\_address\_with\_4k**

Implements Armv8.7 52-bit IPA/PA support for 4k (FEAT\_LPA2).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.7 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: `0`

**has\_56\_bit\_va**

56-bit Physical Address, identified as (FEAT\_LVA3)values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

**has\_64bit\_pmu\_ext\_access**

Implement 64-bit pmu external interface access (FEAT\_PMU\_EXT64)values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.8 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

**has\_64k\_granule**

Implement the 64k LPAE translation granule.

Type: `bool`

Default value: `true`

**has\_aarch32\_dbgdidr\_etc**

DBGDIDR, DBGDRAR, DBGDSAR exist even if EL1 doesn't implement AArch32.

Type: `bool`

Default value: `true`

**has\_aarch32\_hpd**

If true then hierarchical permission disable is supported in AArch32 (FEAT\_AA32HPD).

Type: `bool`

Default value: `false`

**has\_aarch64**

All implemented exception levels can run in AArch64.

Type: `bool`

Default value: `true`

**has\_acp**

If true, Accelerator Coherency Port is configured.

Type: `bool`

Default value: `false`

### **has\_actlr2**

If true ACLTR2 exists and ACTLR2(NS) is aliased to ACTLR\_EL1[63:32].

Type: `bool`

Default value: `false`

### **has\_actlr\_virtualisation**

If true ACTLR\_EL12 is implemented and ACTLR\_EL1 supports virtualisation.

Type: `bool`

Default value: `false`

### **has\_address\_breakpoint\_linking**

Implement Address Breakpoint Linking Extension (FEAT\_ABLE) values of this parameter are:-  
0, feature is not enabled.- 1, feature is implemented if ARMv9.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **has\_afgdt\_virtualisation**

If true AFGDTpn\_EL12 is implemented and AFGDTpn\_EL1 supports virtualisation.

Type: `bool`

Default value: `true`

### **has\_amu**

Implement activity monitor functionality from ARMv8.4 (FEAT\_AMUv1). values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **has\_amu\_ext64**

Implement 64-bit external interface to the Activity Monitors (FEAT\_AMU\_EXT64). values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: `0`

### **`has_amu_extacr`**

If true then implement AMU external access control registers (FEAT\_AMU\_EXTACR).

Type: `bool`

Default value: `false`

### **`has_arm_v8-1`**

Implement the ARMv8.1 Extension.

Type: `bool`

Default value: `false`

### **`has_arm_v8-2`**

Implement the ARMv8.2 Extension.

Type: `bool`

Default value: `false`

### **`has_arm_v8-3`**

Implement the ARMv8.3 Extension.

Type: `bool`

Default value: `false`

### **`has_arm_v8-4`**

Implement the ARMv8.4 Extension.

Type: `bool`

Default value: `false`

### **`has_arm_v8-5`**

Implement the ARMv8.5 Extension.

Type: `bool`

Default value: `false`

### **`has_arm_v8-6`**

Implement the ARMv8.6 Extension.

Type: `bool`

Default value: `false`

### **has\_arm\_v8-7**

Implement the Armv8.7 Extension.

Type: `bool`

Default value: `false`

### **has\_arm\_v8-8**

Implement the ARMv8.8 Extension.

Type: `bool`

Default value: `false`

### **has\_arm\_v8-9**

Implement the ARMv8.9 Extension.

Type: `bool`

Default value: `false`

### **has\_arm\_v9-0**

Implement the ARMv9.0 Extension.

Type: `bool`

Default value: `false`

### **has\_arm\_v9-1**

Implement the ARMv9.1 Extension.

Type: `bool`

Default value: `false`

### **has\_arm\_v9-2**

Implement the ARMv9.2 Extension.

Type: `bool`

Default value: `false`

### **has\_arm\_v9-3**

Implement the ARMv9.3 Extension.

Type: `bool`

Default value: `false`

### **has\_arm\_v9-4**

Implement the ARMv9.4 Extension.

Type: `bool`

Default value: `false`

### **has\_arm\_v9-5**

Implement the ARMv9.5 Extension.

Type: `bool`

Default value: `false`

### **has\_arm\_v9-6**

Implement the ARMv9.6 Extension.

Type: `bool`

Default value: `false`

### **has\_arm\_v9-7**

Implement the ARMv9.7 Extension.

Type: `bool`

Default value: `false`

### **has\_asid2**

If true then support for use of two concurrent ASIDs (FEAT\_ASID2) values of this parameter are:- 1, feature is implemented if ARMv9.5 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_at\_with\_pan**

Implement new AT instructions with PAN support (FEAT\_PAN2). values of this parameter are:- 1, feature is implemented if ARMv8.2 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1



**has\_ats1a**

Support for ATS1ExR instructions (FEAT\_ATS1A) values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

**has\_attribute\_index\_enhancement**

Memory Attribute Index Enhancement (FEAT\_AIE) values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

**has\_axflag\_xaflag**

Implement flag manipulation instructions (AXFlag, XAFlag) from ARMv8.5 (FEAT\_FlagM2). values of this parameter are:- 1, feature is implemented if ARMv8.5 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

**has\_axflag\_xaflag\_frint**

Implement flag manipulation instructions (AXFlag, XAFlag) and floating-point rounding to int instructions (FRINT[32|64][X|Z]) from ARMv8.5. If this parameter is enabled, it also enables both has\_axflag\_xaflag and has\_frint. If support for only one of the features is needed, please use the individual parameters and do not enable this one (FEAT\_FlagM2, FEAT\_FRINTTS). values of this parameter are:- 1, feature is implemented if ARMv8.5 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

**has\_bc**

Implement Armv8.8 Hinted Conditional Branch (FEAT\_HBC) values of this parameter are:- 1, feature is implemented if ARMv8.8 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

**has\_branch\_target\_exception**

Implement Branch target identification mechanism from ARMv8.5 (FEAT\_BTI) (DEPRECATED: Use bti\_support\_level instead). values of this parameter are:- 1, feature is implemented if ARMv8.5 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_brbe**

If true, implements branch record buffer extension (FEAT\_BRBE).

Type: `bool`

Default value: `false`

### **has\_brbe\_v1p1**

If true, implements FEAT\_BRBEv1p1.

Type: `bool`

Default value: `false`

### **has\_ccidx**

Implement the ARMv8.3 FEAT\_CCSDR Extension. Extending the `ccsidr` number of sets (FEAT\_CCIDX).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.3 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **has\_cfinv\_rmif\_setf**

Implement flag manipulation (CFINV, RMIF, SETF8, SETF16) instructions from ARMv8.4 (FEAT\_FlagM).values of this parameter are:- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_cflt**

Support for compare and fault instructions (FEAT\_CFLT)values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.7 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **has\_chkfeat**

Implement CHKFEAT instruction from ARMv9.4 (FEAT\_CHK).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **has\_clear\_bhb**

Implement Clear Branch History information instruction (FEAT\_CLRBHB).values of this parameter are:- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_clear\_other\_speculation\_by\_context**

Implement execution and data prediction invalidation from Armv8.9 (FEAT\_SPECRES2).values of this parameter are:- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_cmh**

Implement Coherency scalability hints (FEAT\_CMH).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.7 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **has\_cmo\_wr\_control**

Whether stage1/2 CMO write perm control is supported (FEAT\_CMOW).values of this parameter are:- 1, feature is implemented if ARMv8.8 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_cmpbr**

Implement compare and branch instructions from ARMv9.6, optional in ARMv9.5 (FEAT\_CMPBR).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.5 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **has\_coherent\_icache**

Whether icache invalidation to the point of unification is required for instruction to data coherence.  
true - Invalidate operations not required.

Type: `bool`

Default value: `false`

### **has\_common\_not\_private\_translations**

Implement the TTBRn\_ELx.CnP (Common not Private) controls from ARMv8.2 (FEAT\_TTCNP).values of this parameter are:- 1, feature is implemented if ARMv8.2 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_complex\_number**

Implement ARMv8.3 complex number support, Multiply Accumulate and Add instructions (FEAT\_FCMA).values of this parameter are:- 1, feature is implemented if ARMv8.3 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_const\_pac**

Feature for singular selection of PAC field (FEAT\_CONSTPACFIELD).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.8 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **has\_cssc**

Support for common short sequence compression instructions (FEAT\_CSSC).values of this parameter are:- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_cvadp\_support**

Implement instruction to support cache clean by deep persistence (DC CVADP) from ARMv8.5, can be selected for core implemented on any arch version starting ARMv8.2 (FEAT\_DPB, FEAT\_DPB2).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.2 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

**has\_data\_abort\_syndrome\_enhancements**

Support for Data Abort syndrome information enhancements (FEAT\_EAESR). values of this parameter are:- 1, feature is implemented if ARMv9.7 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

**has\_data\_alignment\_flag**

Implement non-optimal misalignment flag for PMU/SPE from ARMv8.5. values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.5 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

**has\_debug\_rom**

If true, a debug ROM will be generated describing the cluster's debug components.

Type: `bool`

Default value: `true`

**has\_delayed\_brbe\_records**

If true, a synchronization barrier is required to update the BRB records (FEAT\_BRBE).

Type: `bool`

Default value: `true`

**has\_delayed\_ctireg**

Delay the functional effect of CTI register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**has\_delayed\_mdscr\_el1**

Delay the functional effect of MDSCR\_EL1 register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

### **has\_delayed\_oslar\_el1**

Delay the functional effect of OSLAR\_EL1 register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

### **has\_delayed\_pmureg**

Delay the functional effect of PMU register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

### **has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

### **has\_delayed\_wfe\_trap**

Implements Configurable Delayed WFE trapping from ARMv8.6 (FEAT\_TWED).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.6 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **has\_dgh**

Implements Data Gathering Hint instruction from ARMv8.6 (FEAT\_DGH).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.6 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_dot\_product**

Implement the dot product (UDOT, SDOT) instructions from ARMv8.4 (FEAT\_DotProd).values of this parameter are:- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_e0pd**

Implement ARMv8.5 feature to prevent unprivileged access to one half of the memory (FEAT\_EOPD).values of this parameter are:- 1, feature is implemented if ARMv8.5 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

### **has\_e2h\_rao**

Whether the implementation treats HCR\_EL2.E2H as Read-As-One (**RAO**). 0 : FEAT\_E2H0 implemented.values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.1 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

### **has\_ebep**

Implement Exception-Based Event Profiling from ARMv9.4 (FEAT\_EBEP).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.4 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

### **has\_ebf16**

Support for Extended BFloat16 Behaviours (FEAT\_EBF16).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.2 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

### **has\_ecbhb**

Implement Exploitative Control using Branch History information between exception levels (FEAT\_ECBHB).values of this parameter are:- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

### **has\_edacr**

Implement EDACR register.

Type: `bool`

Default value: `true`

### **has\_edhshr**

Implement external debug halt status register (FEAT\_EDHSHR). 0: FEAT\_EDHSHR is not implemented unless architecturally required by another feature, 1: FEAT\_EDHSHR is implemented, 2: FEAT\_EDHSHR is implemented (extends EDHSHR to include the VNCR, CM, and WnR fields), 0xFF: FEAT\_EDHSHR implementation is dependent on FEAT\_SME.

Type: `uint8_t`

Default value: 15

### **has\_el2**

Implements EL2.

Type: `bool`

Default value: `true`

### **has\_el3**

Implements EL3.

Type: `bool`

Default value: `true`

### **has\_enhanced\_pac**

If pointer authentication is enabled then implement enhanced PAC (FEAT\_EPAC).

Type: `bool`

Default value: `false`

### **has\_enhanced\_pan**

Implements Armv8.7 Enhanced PAN feature (FEAT\_PAN3) values of this parameter are:- 1, feature is implemented if ARMv8.7 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_enhanced\_software\_step**

Implement Enhanced Software Step Extension (FEAT\_STEP2) values of this parameter are:- 1, feature is implemented if ARMv9.5 is enabled.- 2, feature is implemented.

Type: `uint8_t`



Default value: 1

### **has\_ete**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (-plugin or -P).

Type: bool

Default value: false

### **has\_ets**

Whether Enhanced Translation Synchronization is supported. (NOTE: DEPRECATED: use ets\_level instead).

Type: bool

Default value: false

### **has\_exception\_trapping\_form\_of\_vector\_catch**

Implement the exception trapping form of vector catch debug event.

Type: bool

Default value: true

### **has\_extended\_recip\_estimate**

Implements increased precision of reciprocal instructions (FEAT\_RPRES).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.7 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

### **has\_external\_rndr**

Implement external random number generator module. When enabling this with has\_rndr enabled, the external random number generator will be used instead of internal random number generator.values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.5 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

### **has\_f16f32dot**

Implement FEAT\_F16F32DOT.

Type: bool

Default value: `false`

**has\_f16f32mm**

Implement FEAT\_F16F32MM.

Type: `bool`

Default value: `false`

**has\_f16mm**

Implement FEAT\_F16MM.

Type: `bool`

Default value: `false`

**has\_f8f16mm**

Implement FEAT\_F8F16MM and dependent features.

Type: `bool`

Default value: `false`

**has\_f8f32mm**

Implement FEAT\_F8F32MM and dependent features.

Type: `bool`

Default value: `false`

**has\_faminmax**

Implement FEAT\_FAMINMAX.

Type: `bool`

Default value: `false`

**has\_far\_not\_valid**

Implements FnV bit in ESR\_ELx and xFSR, FAR not valid for synchronous external aborts.

Type: `bool`

Default value: `false`

**has\_far\_not\_valid\_dfsc**

Implements FnV bit in ESR\_ELx, FAR not valid for synchronous external aborts for Data Abort.

Type: `bool`

Default value: `false`

### **has\_far\_not\_valid\_ifsc**

Implements FnV bit in ESR\_ELx and xFSR, FAR not valid for synchronous external aborts for Instruction Abort.

Type: `bool`

Default value: `false`

### **has\_fdit**

Support higher exception levels to enforce data-independent timing (FEAT\_FDIT).values of this parameter are:- 1, feature is implemented if ARMv9.7 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_feat\_pops**

Whether ARMv9.6 RAS support for clean-and-invalidate of data by virtual address to Point of Physical Storage (FEAT\_PoPS).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.6 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **has\_fgt**

Implements Fine-grained Virtualization Traps extension from ARMv8.6 (FEAT\_FGT).values of this parameter are:- 1, feature is implemented if ARMv8.6 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_fgt2**

Implement additional FGT traps introduced in ARMv8.9 (FEAT\_FGT2).values of this parameter are:- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_fgwt3**

If true then Fine Grained Write EL3 is enabled (FEAT\_FGWTE3).

Type: `bool`

Default value: `false`

### **has\_fixed\_function\_instr\_counter**

Implement fixed-function instruction counter (FEAT\_PMUv3\_ICNTR) values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **has\_fp16**

Implement the half-precision floating-point data processing instructions from ARMv8.2 (FEAT\_FP16). values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.2 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_fp16\_fmlal**

Implement the New Floating Point Multiplication Variant (FP16 FMLAL, FMLSL) instructions from ARMv8.4. Only supported if `has_fp16=0x1` (FEAT\_FHM). values of this parameter are:- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_fpmr**

Implement FPMR (FEAT\_FPMR).

Type: `bool`

Default value: `false`

### **has\_fprcvt**

Implement FEAT\_FPRCVT FP convert instructions from ARMv9.6, optional in ARMv9.5 (FEAT\_FPRCVT) values of this parameter are:- 1, feature is implemented if ARMv9.6 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

**has\_frint**

Implement floating-point rounding to int instructions (FRINT[32|64][X|Z]) from ARMv8.5 (FEAT\_FRINTTS).values of this parameter are:- 1, feature is implemented if ARMv8.5 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

**has\_gcs**

Implement Guarded Control Stack Extension from ARMv9.4 (FEAT\_GCS).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.4 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

**has\_generic\_authentication**

Implement ARMv8.3 generic authentication.values of this parameter are:- 1, feature is implemented if ARMv8.3 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

**has\_guest\_translation\_granule**

Implement mechanism for guest translation granule identification from ARMv8.5, ID values determined by stage1 granule configuration parameters (FEAT\_GTG).values of this parameter are:- 1, feature is implemented if ARMv8.5 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

**has\_haft**

Implement Hardware managed Access Flag for Table Descriptors (FEAT\_HAFT).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.7 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

**has\_hardware\_accelerator\_for\_cleaning\_dirty\_state**

Whether hardware accelerator for cleaning Dirty state is supported (FEAT\_HACDBS).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.5 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

**has\_hardware\_dirty\_state\_tracking\_structure**

Whether hardware Dirty state tracking Structure is supported (FEAT\_HDBSS).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.5 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

**has\_hardware\_translation\_table\_update**

Type of hardware translation table supported (when enabled by hardware\_translation\_table\_update\_implemented). 0, not implemented. 1, access bit updates implemented. 2, access bit updates and dirty bit mechanism implemented (FEAT\_HAFDBS).

Type: uint8\_t

Default value: 2

**has\_hcrx\_el2**

Implements new HCRX\_EL2 id register from Armv8.7 (FEAT\_HCX).values of this parameter are:- 1, feature is implemented if ARMv8.7 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

**has\_hpmn0**

Allow hypervisor to set MDCR\_EL2.HPMN to 0 (FEAT\_HPMN0).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.8 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

**has\_id\_reg\_read**

Implement read access to the ID registers (ESR\_ELx.EC=0x18) (FEAT\_IDST).values of this parameter are:- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_iesb**

Implement support for implicit error sync event from ARMv8.2 (FEAT\_IESB).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.2 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_internal\_ppu\_support**

If true, in-cluster PPUs are used for V9 cores. If false, Basic power controller/external PPU are used. This parameter is to support software which uses external PPU. When every software migrates to support internal PPU through UtilityBus, this parameter will be removed.

Type: `bool`

Default value: `false`

### **has\_itd**

Implement the optional IT disable feature.

Type: `bool`

Default value: `true`

### **has\_ite**

Implement Instrumentation Trace Extension from ARMv9.4 (FEAT\_ITE).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **has\_jscvt**

Implement ARMv8.3 javascript Floating-point to Integer conversion instruction (FEAT\_JSCVT).values of this parameter are:- 1, feature is implemented if ARMv8.3 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

**has\_large\_system\_ext**

Implement the ARMv8 Large System Extensions (FEAT\_LSE).

Type: `bool`

Default value: `false`

**has\_large\_ttbr\_ba\_without\_lpa**

When FEAT\_LPA is not implemented, whether TTBR base address supports large values (52 bits) or not (48 bits).

Type: `bool`

Default value: `true`

**has\_large\_va**

Implement support for the extended 52-bit virtual addresses from ARMv8.2 (FEAT\_LVA).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.2 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

**has\_ldapur\_stlur**

Implement support for LDAPR and STLUR instructions with immediate offsets from ARMv8.4 (FEAT\_LRCP2).values of this parameter are:- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

**has\_ldm\_stm\_ordering\_control**

Implement the SCTLR\_ELx.LSMAOE (Load/Store Multiple Atomicity and Ordering Enable) and SCTLR\_ELx.nTLSMD (no Trap Load/Store Multiple to Device) controls from ARMv8.2 (FEAT\_LSMAOC).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.2 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

**has\_lorrl**

Support for Limited Order Regions in Realm PA space (FEAT\_LORRL).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.7 is enabled.- 2, feature is implemented.



Type: `uint8_t`

Default value: 0

### **has\_lrcpc**

If true then it support the RCpc feature from ARMv8.3 (FEAT\_LRCPC).

Type: `bool`

Default value: `false`

### **has\_lrcpc3**

Implement Release Consistency processor consistent (RCpc) feature from Armv8.9 (FEAT\_LRCPC3).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **has\_lscp**

Implement 128-bit Load acquire and store release pair single-copy atomic instructions (FEAT\_LSCP)values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.7 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **has\_lsfe**

Implement A64 base Atomic floating-point in-memory instructions (FEAT\_LSFE)values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.3 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **has\_lsui**

Implement additional load and store unprivileged instructions (FEAT\_LSUI).values of this parameter are:- 1, feature is implemented if ARMv9.6 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_lut**

Implement FEAT\_LUT.

Type: `bool`

Default value: `false`

### **has\_mbist\_never1\_ae25**

Implements the MBIST Never1 detection for AE25 class cpu.

Type: `bool`

Default value: `false`

### **has\_mismatch\_and\_range\_breakpoints**

Implement Mismatch and Range Breakpoints (FEAT\_BWE) values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **has\_mismatch\_watchpoints**

Implement Breakpoints and Watchpoints Enhancements (FEAT\_BWE2). values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.5 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **has\_mops\_go\_option**

Implement MTE Tag set only MOPS instructions (FEAT\_MOPS\_GO). 0, not implemented. 1, implemented using Option A. 2, implemented using Option B.

Type: `uint8_t`

Default value: 0

### **has\_mops\_option**

Implement Armv8.8 standard instructions for memory operations (FEAT\_MOPS). 0, not implemented (unsupported if Armv8.8 is enabled). 1, implemented using Option A. 2, implemented using Option B.

Type: `uint8_t`

Default value: 0

**has\_mpam**

Implement ARMv8.4 MPAM Registers and associated functionality (FEAT\_MPAM).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

**has\_mpamv2**

Implement MPAM architecture 2.0 (FEAT\_MPAMv2).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.7 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

**has\_mpamv2\_alt\_id**

Implement alternative identifier for MPAM architecture 2.0 (FEAT\_MPAMv2\_ALT\_ID).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.7 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

**has\_mpamv2\_instr\_alt\_id**

Implement alternative identifier for instruction fetches when FEAT\_MPAMv2\_ALT\_ID is enabled.values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.7 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

**has\_mpamv2\_vid**

Implement identifier virtualization for MPAM architecture 2.0 (FEAT\_MPAMv2\_VID).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.7 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

**has\_mpm**

Implement max-power mitigation mechanism (MPMM).

Type: bool

Default value: `false`

### **has\_mt\_pmu\_disable\_feature**

Implements Multi-threading PMU disable extension from ARMv8.6 (FEAT\_MTPMU). 0: FEAT\_MTPMU is disabled, 1: FEAT\_MTPMU is enabled if ARMv8.6 is implemented, 2: FEAT\_MTPMU is cherry-picked, 0xF: The feature is disabled and is represented by value 0xF in ID\_AA64DFR0\_EL1.MTPMU.

Type: `uint8_t`

Default value: 0

### **has\_mte\_async\_faults**

Whether MTE asynchronous faults are supported (FEAT\_MTE\_ASYNC).

Type: `bool`

Default value: `true`

### **has\_mte\_eirg**

Implement Enhanced Insert Random Tag (FEAT\_MTE\_EIRG) values of this parameter are:- 1, feature is implemented if ARMv9.7 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_mte\_fgt**

When FEAT\_VMTE or FEAT\_MTE2 are implemented, whether FEAT\_MTEFGT is implemented.

Type: `bool`

Default value: `false`

### **has\_mte\_perm**

Implement tag access permission (FEAT\_MTE\_PERM).

Type: `bool`

Default value: `false`

### **has\_mte\_tag\_related\_fault\_high\_prio\_than\_data**

For DC GZVA, Whether MMU faults generated by tag access has higher priority than faults due to data access.

Type: `bool`

Default value: `false`

**has\_nested\_virtualization**

Implement ARMv8.3 nested virtualization (FEAT\_NV).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.3 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

**has\_nmi**

Implement AARCH64 Non-Maskable Interrupts (FEAT\_NMI)values of this parameter are:- 1, feature is implemented if ARMv8.8 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

**has\_no\_os\_double\_lock**

Do not implement the OS double-lock (FEAT\_DoubleLock).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

**has\_non\_context\_synchronizing\_exception\_controls**

Implement cosmetic controls for whether exception entry and exit are context synchronizing events (SCTLR\_ELx.{EIS,EOS}) from ARMv8.5 (FEAT\_ExS).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.5 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

**has\_nv1\_raz**

Whether the implementation treats HCR\_EL2.NV1 as Read-As-Zero (**RAZ**), if has\_e2h\_rao = 1.values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.3 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

**has\_nv\_frac**

Whether the NV\_frac behavior is supported. (DEPRECATED: use nv\_frac\_support\_level)values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

**has\_occmo**

Implement The DC CIVAOC instruction (FEAT\_OCCMO) values of this parameter are:- 1, feature is implemented if ARMv9.6 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

**has\_par\_bit10\_razwi**

Whether PAR\_EL1[10] is **RAZ/WI**.

Type: `bool`

Default value: `false`

**has\_partial\_delayed\_mdscr\_el1**

has\_delayed\_oslar\_el1 only apply to some bits of MDSCR\_EL1 (MDE, KDE, TDCC, SS).

Type: `bool`

Default value: `false`

**has\_pauth\_enhctl**

Support for Enhanced PAC controls (FEAT\_PAuth\_EnhCtl) values of this parameter are:- 1, feature is implemented if ARMv9.7 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

**has\_pc\_sample\_based\_profiling**

If true, pc sample-based profiling is enabled (FEAT\_PCSRv8, FEAT\_PCSRv8p2).

Type: `bool`

Default value: `true`

**has\_pc\_sample\_profiling\_enable**

Whether PC Sample profiling enable is implemented (FEAT\_PCSRv8p9).

Type: `bool`

Default value: `false`

**has\_pcdphint**

Support for producer-consumer data placement hints instructions (FEAT\_PCDPHINT) values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.6 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

**has\_per\_cluster\_debug\_auth\_ports**

If true then the debug authentication ports i.e. `spniden`, `niden`, `rpliden`, `rtpiden`, `dbgen`, `spiden` are available per cluster.

Type: `bool`

Default value: `false`

**has\_peripheral\_port**

If true, an additional AXI peripheral port is configured.

Type: `bool`

Default value: `false`

**has\_permission\_indirection\_s1**

Implement the Permission Indirection Extension at stage 1 (FEAT\_S1PIE) values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

**has\_permission\_indirection\_s2**

Implement the Permission Indirection Extension at stage 2 (FEAT\_S2PIE) values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

**has\_permission\_overlay\_s1**

Implement the Permission Overlay Extension at stage 1 (FEAT\_S1POE) (NOTE: DEPRECATED: Use `s1poe_support_level` instead) values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **has\_permission\_overlay\_s2**

Implement the Permission Overlay Extension at stage 2 (FEAT\_S2POE) values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **has\_plb\_conflict\_abort**

Constrained unpredictable when PLB conflicts observed, True : PLB conflict abort reported.

Type: `bool`

Default value: `false`

### **has\_pmss**

Implement PMU Snapshot Extension from Armv8.9 (FEAT\_PMUv3\_SS) values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **has\_pmu**

Implement the optional Performance Monitors Extension (FEAT\_PMUv3). 0, Not Implemented. 1, Implemented. 2, PMU is IMPLEMENTATION\_DEFINED, PMU version would be set to 0xF and would behave as if no PMU is implemented.

Type: `uint8_t`

Default value: 1

### **has\_pmu\_edge\_detection**

Implement PMU Event edge detection (FEAT\_PMUv3\_EDGE) values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **has\_pmu\_extpmn**

Implement optional PMU extension feature to reserve event counters for external agents from ARMv9.5 (FEAT\_PMUv3\_EXTPMN). 0 not implemented, 1 implemented values of this parameter



are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.5 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

#### **has\_pmu\_for\_sme\_extension**

Implement PMUv3 for Scalable Matrix Extension (SME) from ARMv9.5 (FEAT\_PMUv3\_SME) values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.5 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

#### **has\_pmu\_threshold\_linking\_control**

Implement PMU threshold linking control (FEAT\_PMUv3\_TH2) values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.5 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

#### **has\_pointer\_authentication**

Implement ARMv8.3 pointer authentication (FEAT\_PAuth) values of this parameter are:- 1, feature is implemented if ARMv8.3 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

#### **has\_prediction\_invalidation\_instructions**

Implement execution and data prediction invalidation from ARMv8.5 (FEAT\_SPECRES) values of this parameter are:- 1, feature is implemented if ARMv8.5 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

#### **has\_prfm\_slc**

Implement PRFM with SLC hint (FEAT\_PRFM\_SLC) values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

**has\_pstate\_dit**

Implement timing insensitivity of data processing instructions from ARMv8.4 (FEAT\_DIT).values of this parameter are:- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

**has\_pstate\_pan**

Implement the PSTATE.PAN (Privileged Access Never) control from ARMv8.1 (FEAT\_PAN).values of this parameter are:- 1, feature is implemented if ARMv8.1 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

**has\_pstate\_uao**

Implement the PSTATE.UAO (User Access Override) control from ARMv8.2 (FEAT\_UAO).values of this parameter are:- 1, feature is implemented if ARMv8.2 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

**has\_qarma3\_pac**

Supports QARMA3 pointer authentication algorithm (FEAT\_PACQARMA3).

Type: `bool`

Default value: `false`

**has\_ras**

Implements the ARMv8 RAS Extension. 0 = NO\_RAS, 1 = MINIMAL\_RAS, 2 = FULL\_RAS (FEAT\_RAS).

Type: `uint8_t`

Default value: 0

**has\_ras\_aderr**

Implement RAS Asynchronous Device Read Error from Armv8.9 (FEAT\_ADERR).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

**has\_ras\_anerr**

Implement RAS Asynchronous Normal Read Error from Armv8.9 (FEAT\_ANERR).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

**has\_ras\_armv84\_extension**

Implement ARMv8.4 RAS Extension (FEAT\_RASv1p1).values of this parameter are:- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

**has\_ras\_armv89\_double\_fault**

Implement RAS Double Fault Extension from Armv8.9 (FEAT\_DoubleFault2).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

**has\_ras\_armv89\_extension**

Implement RAS extension from Armv8.9 (FEAT\_RASv2).values of this parameter are:- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

**has\_ras\_critical\_error**

[DEPRECATED: Set CI field on first register in error\_record\_feature\_register JSON instead]

ARMv8.4 AArch64 RAS Critical Error is implemented or not. 0 - Feature Not Supported, 1 - Feature always enabled, 2 - Feature is controllable.

Type: uint8\_t

Default value: 0

**has\_ras\_delegated\_serror\_exceptions\_for\_el3**

Implement Delegated SError exceptions for EL3 (FEAT\_E3DSE).values of this parameter are:- 1, feature is implemented if ARMv9.5 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

### **has\_ras\_double\_fault**

Implement ARMv8.4 RAS Double Fault Extension (FEAT\_DoubleFault).values of this parameter are:- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

### **has\_ras\_fault\_injection**

[DEPRECATED: Set INJ field on first register in error\_record\_feature\_register JSON instead]

Implement ARMv8.4 Standard Fault Injection mechanism.values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

### **has\_ras\_mmmap\_view**

Implement memory mapped view of RAS Registers for cores.

Type: bool

Default value: false

### **has\_ras\_pfar**

Implement RAS Physical Fault Address Registers from Armv8.9 (FEAT\_PFAR).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

### **has\_ras\_timestamp**

[DEPRECATED: Set TS field on first register in error\_record\_feature\_register JSON instead]

ARMv8.4 AArch64 RAS Timestamp register is implemented or not. 0 - No Timestamp is recorded, 1 - Generic Timer timestamp is recorded, 2 - IMP DEF timestamp is recorded.

Type: uint8\_t

Default value: 0

### **has\_rassa\_acr**

Implement RAS System Architecture v2 optional access control register (FEAT\_RASSA\_ACR).

Type: bool

Default value: `false`

### **has\_restriction\_on\_speculative\_data\_loaded**

Implements the ARMv8.5 security feature (Restrictions on the effects of speculation) (FEAT\_CSV3).values of this parameter are:- 1, feature is implemented if ARMv8.5 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_rme**

If true, implements full realm management extension (FEAT\_RME). Note: This parameter is deprecated and will be removed in future releases, please use `rme_support_level` parameter.

Type: `bool`

Default value: `false`

### **has\_rme\_gdi**

Support for RME granular data isolation (FEAT\_RME\_GDI).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.6 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **has\_rme\_gpc2**

If true then RME GPC2 extension is enabled (FEAT\_RME\_GPC2).

Type: `bool`

Default value: `false`

### **has\_rme\_gpc3**

If true then RME GPC3 extension is enabled (FEAT\_RME\_GPC3).

Type: `bool`

Default value: `false`

### **has\_rndr**

Implement random number instructions to read from RNDR and RNDRSS random number registers from ARMv8.5 (FEAT\_RNG).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.5 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **has\_rndr\_trap**

Implement trapping for RNDR and RNDRSS random number registers from ARMv8.8. (FEAT\_RNG\_TRAP) values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.8 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **has\_rounding\_doubling\_multiply\_add\_subtract**

Implement the rounding doubling multiply add and subtract instructions from ARMv8.1 (FEAT\_RDM). values of this parameter are:- 1, feature is implemented if ARMv8.1 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_rprfm**

Support for RPRFM hint instruction (FEAT\_RPRFM).

Type: `bool`

Default value: `false`

### **has\_sbistc\_ae25**

Implements the SBISTC for AE25 class cpu.

Type: `bool`

Default value: `false`

### **has\_scr2**

Support for SCR2\_EL3 register (FEAT\_SCR2) values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.7 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **has\_sctlr2**

Implement SCTL2\_ELx registers (FEAT\_SCTL2) values of this parameter are:- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_sebep**

Implement Synchronous-Exception-Based Event Profiling from ARMv9.4 (FEAT\_SEBEP).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **has\_secure\_el2**

Implement support for Secure EL2 (FEAT\_SEL2).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_self\_hosted\_trace\_extension**

Implement support for the Self-hosted Trace Extensions from ARMv8.4 (FEAT\_TRF).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_small\_page\_table**

Implement small page table support which increases the maximum value of TxSZ field from ARMv8.4 (FEAT\_TTST). Note: will be unimplemented only if both `has_small_page_table=0x0` and `has_secure_el2=0x0`.values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_software\_lock**

Implement software lock in memory-mapped CTI, PMU, and external debug interfaces.

Type: `bool`

Default value: `true`

### **has\_spe\_eft**

Implement SPE extended operation type filtering from ARMv9.5 (FEAT\_SPE\_EFT).values of this parameter are:- 1, feature is implemented if ARMv9.5 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_spe\_fds**

Implement SPE filter by data source from ARMv8.9 (FEAT\_SPE\_FDS) values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **has\_spe\_for\_sme\_extension**

Implement support of SME to SPE from ARMv9.5 (FEAT\_SPE\_SME) values of this parameter are:- 1, feature is implemented if ARMv9.5 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_spe\_fpf**

Implement SPE operation type extension for ASIMD and FP from ARMv9.5 (FEAT\_SPE\_FPF) values of this parameter are:- 1, feature is implemented if ARMv9.5 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_spe\_nvm**

Implement Statistical Profiling physical address mode (FEAT\_SPE\_nVM) values of this parameter are:- 1, feature is implemented if ARMv9.6 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_speculation\_barrier\_inst**

Implement speculation barrier instruction (SB) from ARMv8.5 (FEAT\_SB) values of this parameter are:- 1, feature is implemented if ARMv8.5 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_speculative\_sei**

If true, the PE can generate SError interrupt exceptions from speculative reads of memory, including speculative instruction fetches.



Type: `bool`

Default value: `false`

### **has\_srmask**

Implement bitwise write masks for EL1 control registers (FEAT\_SRMASK).values of this parameter are:- 1, feature is implemented if ARMv9.6 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_srmask2**

Implement bitwise write masks for HCR(X)EL2 registers (FEAT\_SRMASK2).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.7 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **has\_stage2\_ap\_speculative\_update**

Speculative update of S2 AP bit on S1 TTW. 0 = No Update, 1 = Update, 2 = Update including for AT ops.

Type: `uint8_t`

Default value: 0

### **has\_stage2\_fwb**

Implement HCR\_EL2.FWB, stage 2 control of memory types and cacheability (FEAT\_S2FWB).values of this parameter are:- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_stage2\_xnx**

Implement the extended XN[1:0] stage 2 control from ARMv8.2 (FEAT\_XNX).values of this parameter are:- 1, feature is implemented if ARMv8.2 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_stage2\_xnx\_in\_aarch32**

Implement the extended XN[1:0] stage 2 control from ARMv8.2 for Aarch32 (FEAT\_XNX).

Type: `bool`

Default value: `true`

### **has\_statistical\_profiling**

Whether Statistical Based Profiling is implemented (FEAT\_SPE).

Type: `bool`

Default value: `true`

### **has\_supersections**

Whether VMSAv8-32 supersection to support more than 32-bit PA using short descriptor is implemented.

Type: `bool`

Default value: `true`

### **has\_sve**

Whether SVE is implemented (FEAT\_SVE). Note: this is required to enable SME (FEAT\_SME) with `sve.has_sme=1`. An SME only implementation can be enabled by setting both as well as `sve.sme_only=1`.

Type: `uint8_t`

Default value: `0`

### **has\_synchronous\_load\_atomics**

Report asynchronous abort due to unsupported load atomics as synchronous (Cacheable).

Type: `bool`

Default value: `true`

### **has\_synchronous\_load\_atomics\_noncacheable**

Report asynchronous abort due to unsupported load atomics as synchronous (Non-Cacheable).

Type: `bool`

Default value: `true`

### **has\_synchronous\_store\_atomics**

Report asynchronous abort due to unsupported store atomics as synchronous (Cacheable).

Type: `bool`

Default value: `false`

**has\_synchronous\_store\_atomics\_noncacheable**

Report asynchronous abort due to unsupported store atomics as synchronous (Non-Cacheable).

Type: `bool`

Default value: `false`

**has\_sysinstr128**

Support for System Instructions that can take 128-bit inputs (FEAT\_SYSINSTR128) values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

**has\_sysreg128**

Support for 128-bit System Registers (FEAT\_SYSREG128) values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

**has\_tag\_cache\_operations**

Implement tag cache operations DC ZGBVA and DC GBVA (FEAT\_MTETC) values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.6 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

**has\_tcr2**

Implement TCR2\_ELx registers (FEAT\_TCR2) values of this parameter are:- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

**has\_tev**

Support for exception-like overlay management mechanism (FEAT\_TEV) values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if Future Architecture Technologies (FAT) is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **has\_tidcp1**

Implement Armv8.8 ELO use of implementation defined functionality (FEAT\_TIDCP1) values of this parameter are:- 1, feature is implemented if ARMv8.8 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_tlb\_conflict\_abort**

Detected inconsistent TLB content generate aborts. Ignored if FEAT\_BBML3 is implemented.

Type: `bool`

Default value: `false`

### **has\_tlb\_pa\_caching**

Whether intermediate caching of translation table walks might include NonCoherent caches of previous valid walks. 0, NonCoherent caches might be included. 1, No NonCoherent caches included (FEAT\_nTLBPA).

Type: `bool`

Default value: `false`

### **has\_tlbi\_range**

Implement support for TLB Range Maintenance instructions (TLBI RVAE1, etc) from ARMv8.4 (FEAT\_TLBIRANGE). values of this parameter are:- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_tlbi\_to\_outer\_shareable**

Implement support for TLB Maintenance instructions that extend to the Outer Shareable domain (TLBI VAE1OS, etc) from ARMv8.4 (FEAT\_TLBIOS). values of this parameter are:- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_tlbi\_ttl**

Implement support for the TTL level hint in by-address TLB Maintenance instructions from ARMv8.4 (FEAT\_TTL). values of this parameter are:- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_tlbld**

Whether TLBIDomains are implemented in this model. (FEAT\_TLBID)values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.7 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **has\_tlbw**

Implement TLBI instruction for stage2 dirty (FEAT\_TLBW).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.5 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **has\_translation\_hardenig**

Implement the Translation Hardening Extension (FEAT\_THE)values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **has\_trbe**

If true, implements the Trace Buffer Extension (FEAT\_TRBE).

Type: `bool`

Default value: `false`

### **has\_trbe\_ext**

Implements the Trace Buffer external mode extension (FEAT\_TRBE\_EXT).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **has\_trc\_ext**

If true, Allow DAP accesses to Trace registers(FEAT\_TRC\_EXT).

Type: `bool`

Default value: `true`

### **has\_uinj**

Implement software injection of Undefined Instruction exceptions (FEAT\_UINJ).values of this parameter are:- 1, feature is implemented if ARMv9.6 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_unaligned\_single\_copy\_atomicity**

Implement support for SCTL<sub>R</sub>.EL<sub>x</sub>.nAA from ARMv8.4, and A64 atomic, exclusive and acquire/release instructions accessing unaligned bytes inside a 16byte window will not generate alignment fault (FEAT\_LSE2).values of this parameter are:- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_unsupported\_exclusive\_fault**

Report unsupported exclusive access with Unsupported Exclusive fault status (otherwise use external abort).

Type: `bool`

Default value: `true`

### **has\_v8\_4\_debug\_extension**

Implement ARMv8.4 debug extensions (FEAT\_Debugv8p4).values of this parameter are:- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_v8\_4\_pmu\_extension**

Implement PMU extension from ARMv8.4 (FEAT\_PMUv3p4).values of this parameter are:- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

**has\_v8\_5\_debug\_over\_power\_down**

Implement ARMv8.5 Debug over powerdown (FEAT\_DoPD).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.5 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

**has\_v8\_5\_pmu\_extension**

Implement PMU extension from ARMv8.5 (FEAT\_PMUv3p5).values of this parameter are:- 1, feature is implemented if ARMv8.5 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

**has\_v8\_5\_spe\_extension**

Implement SPE extension from ARMv8.5 (FEAT\_SPEv1p1).values of this parameter are:- 1, feature is implemented if ARMv8.5 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

**has\_v8\_6\_pmu\_events**

Implements PMU events from ARMv8.6.values of this parameter are:- 1, feature is implemented if ARMv8.6 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

**has\_v8\_7\_fp\_enhancements**

Implements the Floating Point enhancements from Armv8.7 (introduces FPCR.FIZ/AH/NEP, etc. (FEAT\_AFP).)values of this parameter are:- 1, feature is implemented if ARMv8.7 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

**has\_v8\_7\_pmu\_events**

Implement PMU events from ARMv8.7.values of this parameter are:- 1, feature is implemented if ARMv8.7 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

#### **has\_v8\_7\_pmu\_extension**

Implement PMU extension from ARMv8.7 (FEAT\_PMUv3p7).values of this parameter are:- 1, feature is implemented if ARMv8.7 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

#### **has\_v8\_7\_spe\_extension**

Implement SPE extension from ARMv8.7 (FEAT\_SPEv1p2)values of this parameter are:- 1, feature is implemented if ARMv8.7 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

#### **has\_v8\_7\_spe\_inverted\_filtering**

Where FEAT\_SPEv1p2 is implemented, whether inverted filtering by events is implemented (represented by PMISDR.FnE).

Type: `bool`

Default value: `true`

#### **has\_v8\_7\_spe\_previous\_branch\_target**

Where FEAT\_SPEv1p2 is implemented, whether the optional branch target feature is implemented (FEAT\_SPE\_PBT).

Type: `bool`

Default value: `true`

#### **has\_v8\_8\_debug\_extension**

Implement ARMv8.8 debug extensions (FEAT\_Debugv8p8)values of this parameter are:- 1, feature is implemented if ARMv8.8 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

#### **has\_v8\_8\_pmu\_events**

Implement PMU events from ARMv8.8 (FEAT\_PMUv3).values of this parameter are:- 1, feature is implemented if ARMv8.8 is enabled.- 2, feature is implemented.

Type: `uint8_t`



Default value: 1

#### **has\_v8\_8\_pmu\_extension**

Implement PMU extension from ARMv8.8 (FEAT\_PMUv3p8).values of this parameter are:- 1, feature is implemented if ARMv8.8 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

#### **has\_v8\_8\_spe\_extension**

Implement SPE extension from ARMv8.8 (FEAT\_SPEv1p3).values of this parameter are:- 1, feature is implemented if ARMv8.8 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

#### **has\_v8\_9\_debug\_extension**

Implement ARMv8.9 debug extensions (FEAT\_Debugv8p9).values of this parameter are:- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

#### **has\_v8\_9\_pc\_sample\_based\_profiling**

Implement PC Sample-based Profiling from ARMv8.9 (FEAT\_PCSRv8p9).values of this parameter are:- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

#### **has\_v8\_9\_pmu\_events**

Implement PMU events from ARMv8.9 (FEAT\_PMUv3).values of this parameter are:- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

#### **has\_v8\_9\_pmu\_extension**

Implement PMU extension from ARMv8.9 (FEAT\_PMUv3p9).values of this parameter are:- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

### **has\_v8\_9\_spe\_extension**

Implement SPE extension from ARMv8.9 (FEAT\_SPEv1p4) values of this parameter are:- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

### **has\_v9\_6\_spe\_extension**

Implement FEAT\_SPEv1p5 and FEAT\_SPE\_EXC from ARMv9.6 values of this parameter are:- 1, feature is implemented if ARMv9.6 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

### **has\_virtual\_tag\_check\_on\_load**

When FEAT\_VMTETC is implemented specify whether FEAT\_VMTETCL is implemented. Please note that in order to enable FEAT\_VMTETCL with this parameter, vmte\_support\_level must be set to 1 or 2.

Type: bool

Default value: true

### **has\_vnchr\_el2**

Implement support for nested virtualization enhancements from ARMv8.4 (FEAT\_NV2) values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

### **has\_wfet\_and\_wfit**

Implements WFE and WFI with Timeout from Armv8.7 (FEAT\_WFxT) values of this parameter are:- 1, feature is implemented if ARMv8.7 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

### **has\_writebuffer**

Implement write accesses buffering before L1 cache. May affect ext\_abort behaviour.

Type: bool

Default value: `false`

### **has\_xs**

Implements Armv8.7 XS, TLBInXS, DSBnXS instruction (FEAT\_XS).values of this parameter are:- 1, feature is implemented if ARMv8.7 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **hcptr\_tta\_behaviour**

Behaviour of HCPTR.TTA when there is no CP14 ETM interface. 0, **RAZ/WI**. 1, **RAO/WI**. 2, stateful.

Type: `uint8_t`

Default value: 2

### **hcr\_el2\_miocnce\_is\_rw**

If true, HCR\_EL2.MIOCNCE is treated as R/W instead of **RAZ/WI**; always set to false in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **hcr\_swio\_res1**

Whether HCR.SWIO and/or HCR\_EL2.SWIO are **RES1**.

Type: `bool`

Default value: `false`

### **hdbss\_error\_fault\_type**

Type of fault reported for HDBSS errors. 0 = precise exception, 1 = fault logged in HDBSSPROD\_EL2.FSC (FEAT\_HDBSS).

Type: `uint8_t`

Default value: 0

### **hpfar\_unknown\_when\_ipa\_invalid**

If true, HPFAR\_EL2 is set to 0 when IPA is not valid for stage 2 faults.

Type: `bool`

Default value: `false`

**hpfar\_update\_behaviour**

Defines HPFAR\_EL2 update condition. 0: Always updated on faults taken to EL2. 1: Only when IPA is valid. 2: When IPA is valid or unknown.

Type: `uint8_t`

Default value: 0

**hsr\_uncond\_cc**

Condition codes reported in HSR as AL if it passes.

Type: `bool`

Default value: `false`

**hvbar\_reset\_is\_rvbar**

If true then the reset value of HVBAR is RVBAR, if false the reset value is **UNKNOWN**.

Type: `bool`

Default value: `false`

**icache-hit\_latency**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**icache-log2linelen**

If nonzero, Log2 of the instruction cache line length in bytes (valid values in range 4-8). Otherwise the value of `cache-log2linelen` is used.

Type: `uint8_t`

Default value: 0

**icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-nprefetch**

Number of next sequential instruction cache lines to prefetch. This is only used when `icache-prefetch_enabled=true`.

Type: `uint32_t`

Default value: `0x1`

**icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

**icache-prefetch\_level**

0 based cache level at which instructions are pre-fetched. This is only used when `icache-prefetch_enabled=true`.

Type: `uint8_t`

Default value: `0`

**icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-read\_bus\_width\_in\_bytes**

L1 I-Cache read bus width in bytes used to calculate per-access timing annotations.

Type: `uint32_t`

Default value: 0x8

### **icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **icache-size**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: 0x8000

### **icache-state\_modelled**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **icache-ways**

L1 I-Cache number of ways (sets are implicit from size).

Type: `uint8_t`

Default value: 2

### **id\_spec\_fpacc\_raz**

If true, implementation opts not to disclose the speculative use of pointers processed by a PAC authentication failure by having value 0 for `Spec_FPACC` bits of `ID_AA64MMFR3_EL1` register.

Type: `bool`

Default value: `false`

### **idte3\_support\_level**

Support for trapping ID register accesses to EL3 (FEAT\_IDTE3): 0 - Not implemented. 1 - Implemented.

Type: `uint8_t`

Default value: 0

**iesb\_use\_pre\_cse\_ctx**

If true, read SCTLR\_ELx.IESB and SCR\_EL3.{NMEA,EA} before synchronizing registers. The pre-synchronization values will be used to determine whether to take an implicit ESB.

Type: `bool`

Default value: `false`

**ignore\_DBGPRCR\_CWRR**

Ignore writes to the deprecated DBGPRCR.CWRR bit.

Type: `bool`

Default value: `false`

**ignore\_access\_flag\_update\_by\_CMOS**

If true, CMOS(cache maintenance operations) neither update the access flag nor generate access flag fault.

Type: `bool`

Default value: `false`

**ignore\_access\_flag\_update\_by\_at\_ops**

If true, AT operations do not update access flag.

Type: `bool`

Default value: `false`

**ignore\_data\_abt\_on\_af\_update\_by\_at\_ops**

If true, Data abort generated on AF update by AT operations are ignored. This parameter is only valid if `ignore_access_flag_update_by_at_ops` is false.

Type: `bool`

Default value: `true`

**ignore\_large\_address\_top\_bits\_in\_page\_walk**

Whether page table bits [15:12] are ignored if PA\_SIZE < 52 and output address is configured < 52 with large page.

Type: `bool`

Default value: `false`

**ignore\_tag\_check\_dcc\_load\_store\_in\_ma\_mode\_when\_tco\_is\_disabled**

Constrained unpredictable behavior for reads/writes to external debug interface DTR regs in memory access mode when PSTATE.TCO is 0. If true, tag check is ignored else, tag check is performed if required.

Type: `bool`

Default value: `false`

**ignore\_traps\_to\_dcc\_regs\_in\_debug**

Whether traps get ignored for the following registers in debug state: AArch64: MDCCSR\_EL0, OSDTREG\_EL1, OSDTRTX\_EL1, MDCCINT\_EL1. AArch32: DBGDSCRint, DBGDIDR, DBGDSAR, DBGDRAR,, DBGDTRTXext, DBGDCCINT.

Type: `bool`

Default value: `false`

**illegal\_state\_exception\_priority**

IMPDEF priority of Illegal State Exception. 0: After breakpoint exceptions 1: Before Instruction Abort.

Type: `uint8_t`

Default value: 0

**imp\_def\_functionality\_behaviour**

Behaviour of **IMPLEMENTATION DEFINED** registers and system instructions. 0, UNDEF. 1, **RAZ/WI**.

Type: `uint8_t`

Default value: 0

**impdef\_regs\_and\_unpred\_from\_implementation**

Configure implementation defined registers and unpredictable behaviour to match the specified implementation. Requires a license for the selected implementation model. User has to provide the default values for the published or configurable parameters through commandline arguments. Use ARM\_Cortex-A<num> or ARM\_<codename> for licensed pre-release cores.

Type: `string`

Default value: `N/A`

**impdef\_sysreg\_json**

Configure mask/reset for impdef registers in a JSON format. a string (max 1024 chars) or a filename, starting with '@' sign. is a list of objects, with following attributes:.

Type: `string`



Default value: "[ ]"

### **independent\_cache\_control\_traps**

Implement Independent Cache Control traps from ARMv8.5. 0, No support. 1, Supported but not for tlb maintenance instructions. 2, Full support. (FEAT\_EVT).

Type: `uint8_t`

Default value: 0

### **inner\_cache\_boundary**

CLIDR.ICB, cache level boundary between inner and outer shareable domains.

Type: `uint8_t`

Default value: 0

### **insert\_iesb\_before\_exception**

If true then inserts an IESB before taking with Exception otherwise has no effect and IESB is taken after PState is changed due to the Exception.

Type: `bool`

Default value: `false`

### **instruction\_tlb\_size**

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

Type: `uint32_t`

Default value: 0x0

### **internal\_vgic**

Instantiate VGIC peripheral in this processor.

Type: `bool`

Default value: `false`

### **invalidate\_code\_cache\_on\_icache\_cmo**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: `uint8_t`

Default value: 0

**irt\_fetch\_fault\_on\_data\_check\_report\_type**

Behaviour for IRT fetch faults on data permission checks (can only happen for non-standard data accesses): 0, report as data abort. 1, report as instruction abort.

Type: `bool`

Default value: `false`

**is\_debug\_state\_pmu\_snapshot\_allowed**

If true, PMU snapshot is allowed in debug state.

Type: `bool`

Default value: `true`

**is\_first\_pcsr\_sample\_ignored**

If true, First read of PMPCSR register after reset returns 0xFFFFFFFF.

Type: `bool`

Default value: `false`

**is\_mt\_res0**

If ARMv8.6 is not implemented, and PMUv3 is implemented, this parameter controls whether PMEVTYPER<n>.MT bit is **RES0** or RW. For other implementations, this parameter has no effect.

Type: `bool`

Default value: `false`

**is\_ras\_irq\_edge\_triggered**

If true, ras interrupt is edge-triggered. Otherwise, it's level-triggered.

Type: `bool`

Default value: `true`

**is\_serror\_edge\_triggered**

If true, SError is edge-triggered. Otherwise, it's level-triggered.

Type: `bool`

Default value: `true`

**is\_tagged\_nsh\_treated\_as\_tagged**

Whether a tagged NonShared memory attribute is treated as tagged or not.

Type: `bool`

Default value: `true`

### **`is_uniprocessor`**

Value for the U bit in MPIDR. `true` disables L1 cache coherency protocols.

Type: `bool`

Default value: `false`

### **`isb_is_branch`**

If `true`, ISB is considered an immediate branch. This allows to count ISB as a branch in BRBE.

Type: `bool`

Default value: `false`

### **`ish_is_osh`**

Whether Innershareable is same as OuterShareable.

Type: `bool`

Default value: `false`

### **`itd_conditional_instructions_are_32bit`**

When `SCTLR_ELx.ITD=1`, an IT instruction plus a T16 instruction are considered a single 32bit conditional instruction.

Type: `bool`

Default value: `false`

### **`jidr_is_undef_at_el0`**

If `true`, JIDR register access is UNDEF at EL0.

Type: `bool`

Default value: `false`

### **`jmcr_is_undef_at_el0`**

If `true`, JMCR register access is UNDEF at EL0.

Type: `bool`

Default value: `false`

**joscr\_is\_undef\_at\_el0**

If true, JOSCR register access is UNDEF at EL0.

Type: `bool`

Default value: `false`

**l3\_dataram\_ecc\_granule**

L3 Data RAM ECC granule. Allowed values are 128 or 256 bits.

Type: `uint16_t`

Default value: 128

**l3\_dataram\_read\_latency**

L3 data ram read latency. where DSU allows two or three cycles output delay from L3 data RAM.

Type: `uint8_t`

Default value: 2

**l3\_dataram\_reg\_slice**

L3 data ram read register slice present if true, not present otherwise.

Type: `bool`

Default value: `false`

**l3\_dataram\_write\_latency**

L3 data ram write latency. where DSU allows one, two or three (two cycle latency plus one cycle hold) input delay.

Type: `uint8_t`

Default value: 1

**l3cache-has\_mpam**

L3 Cache has MPAM support.

Type: `bool`

Default value: `false`

**l3cache-hit\_latency**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **l3cache-maintenance\_latency**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **l3cache-miss\_latency**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **l3cache-mpamf.arch\_major\_ver**

L3 Cache MPAMF\_AIDR architecture major version.

Type: `uint8_t`

Default value: `0`

### **l3cache-mpamf.arch\_minor\_ver**

L3 Cache MPAMF\_AIDR architecture minor version.

Type: `uint8_t`

Default value: `0`

### **l3cache-mpamf.esr\_mask**

L3 Cache MPAMF\_ESR mask value.

Type: `uint32_t`

Default value: `0xffffffff`

### **l3cache-mpamf.has\_esr**

L3 Cache's MPAMF\_ESR, MPAMF\_ECR, and MPAM error handling implemented.

Type: `bool`

Default value: `false`

**l3cache-mpamf.has\_extd\_esr**

L3 Cache's MPAMF\_ESR is 64-bits.

Type: `bool`

Default value: `false`

**l3cache-mpamf.has\_impl\_idr**

L3 Cache's MPAMF\_IMPL\_IDR is present.

Type: `bool`

Default value: `false`

**l3cache-mpamf.has\_mbwu\_long\_counter**

L3 Cache has long MBWU counter and capture registers.

Type: `bool`

Default value: `false`

**l3cache-mpamf.has\_mpamfidr\_ext**

MPAMF\_IDR.EXT support.

Type: `bool`

Default value: `false`

**l3cache-mpamf.has\_partid\_nrw**

Narrowing part ID register is present. This is global rather than per-instance.

Type: `bool`

Default value: `false`

**l3cache-mpamf.has\_priority\_partitioning**

The selected resource has priority partitioning described in MPAMF\_PRI\_IDR.

Type: `bool`

Default value: `false`

**l3cache-mpamf.has\_prod\_id**

L3 Cache MPAMF\_IIDR product ID supported.

Type: `uint16_t`

Default value: `0x0`

**l3cache-mpamf.has\_prod\_rev**

L3 Cache MPAMF\_IIDR product REVISION supported.

Type: `uint8_t`

Default value: 0

**l3cache-mpamf.has\_prod\_var**

L3 Cache MPAMF\_IIDR product VARIENT supported.

Type: `uint8_t`

Default value: 0

**l3cache-mpamf.has\_ris**

L3 Cache has resource instance selection support.

Type: `bool`

Default value: `false`

**l3cache-mpamf.max\_partid\_ns**

Maximum value of non-secure PARTID supported for DSU L3Cache. Options are 7 or 63.

Type: `uint8_t`

Default value: 63

**l3cache-mpamf.max\_partid\_rl**

L3 Cache Maximum value of realm PARTID supported for RME implementations.

Type: `uint16_t`

Default value: `0xffff`

**l3cache-mpamf.max\_partid\_rt**

L3 Cache Maximum value of root PARTID supported for RME implementations.

Type: `uint16_t`

Default value: `0xffff`

**l3cache-mpamf.max\_partid\_s**

L3 Cache Maximum value of secure PARTID supported.

Type: `uint16_t`

Default value: `0xffff`

**l3cache-mpamf.max\_pmg\_ns**

L3 Cache Maximum value of non-secure PMG supported.

Type: `uint8_t`

Default value: 255

**l3cache-mpamf.max\_pmg\_r1**

L3 Cache Maximum value of realm PMG supported for RME implementations.

Type: `uint8_t`

Default value: 255

**l3cache-mpamf.max\_pmg\_rt**

L3 Cache Maximum value of root PMG supported for RME implementations.

Type: `uint8_t`

Default value: 255

**l3cache-mpamf.max\_pmg\_s**

L3 Cache Maximum value of secure PMG supported.

Type: `uint8_t`

Default value: 255

**l3cache-mpamf.mbwu\_long\_counter\_width**

L3 Cache long MBWU counter width. 0: 63 bits, 1: 44 bits.

Type: `uint8_t`

Default value: 0

**l3cache-mpamf.no\_impl\_msmon**

L3 Cache's MPAMF\_IMPL\_IDR does not describe resource monitors.

Type: `bool`

Default value: `false`

**l3cache-mpamf.no\_impl\_part**

L3 Cache's MPAMF\_IMPL\_IDR does not describe resource partitioning controls.

Type: `bool`

Default value: `false`



**l3cache-mpamf.ris\_max**

L3 Cache's largest resource instance selector value defined.

Type: `uint8_t`

Default value: 0

**l3cache-mpamf\_base**

L3 Cache memory mapped MPAM registers base address.

Type: `uint64_t`

Default value: 0x0

**l3cache-read\_access\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size `l3cache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**l3cache-read\_bus\_width\_in\_bytes**

L3 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: `uint32_t`

Default value: 0x8

**l3cache-read\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l3cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**l3cache-size**

L3 Cache size in bytes.

Type: `uint32_t`

Default value: 0x0

**l3cache-snoop\_data\_transfer\_latency**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-snoop\_issue\_latency**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-ways**

L3 Cache number of ways (sets are implicit from size).

Type: `uint16_t`

Default value: `16`

**l3cache-write\_access\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size `l3cache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-write\_bus\_width\_in\_bytes**

L3 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: `uint32_t`

Default value: `0x8`

**l3cache-write\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l3cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**legacy\_combining\_exc\_catch\_trace**

Whether exception catch is traced as part of exception entry/exit in same cycle.

Type: `bool`

Default value: `true`

**limit\_ete\_revision\_without\_rme**

If true, limit the ETE revision to ETEv1p1 when FEAT\_RME is not implemented.

Type: `bool`

Default value: `false`

**log2-num\_sf\_index**

Log2 of the number of snoop filter indexes.

Type: `uint8_t`

Default value: 1

**log2\_trace\_buffer\_alignment**

Log2 of trace buffer alignment constraint for output buffer (0->1B ... 11->2Kib).

Type: `uint8_t`

Default value: 0

**ls64\_ignore\_s1\_unpred\_memattr\_transformation**

If true, stage 1 unpredictable memory attribute transformations are ignored for FEAT\_LS64 single-copy atomic 64-byte load/store instructions' (FEAT\_LS64, FEAT\_LS64\_V, FEAT\_LS64\_ACCDATA).

Type: `bool`

Default value: `false`

**ls64\_memtype\_check\_use\_combined\_memattr**

FEAT\_LS64 single-copy atomic 64-byte load/store instructions' 0 : memory attributes check is performed at each enabled stage of translation, 1 : memory attributes check is done on the combined memory attributes only. 2. memory attributes check is done on the combined memory attributes with Stage1 and Stage2 fault get evaluated to check on which stage fault should be reported.

Type: `uint8_t`

Default value: 0

**ls64wb\_memtype\_check\_allow\_any\_cacheable\_memattr**

If true, when FEAT\_LS64WB is implemented, any cacheable memory access performed by LD/ST64B instructions is 64-byte, single-copy atomic.

Type: `bool`

Default value: `false`

**mask\_trbtrg\_res0**

If true, and TRBIDR\_EL1.Align>0, treat TRBTRG\_EL1[TRBIDR\_EL1.Align-1:0] as **RES0** for writes.

Type: `bool`

Default value: `false`

**master\_interface**

Main master interface, CHI if true, AXI otherwise.

Type: `bool`

Default value: `false`

**max\_32bit\_el**

Maximum exception level supporting AArch32 modes. -1: No Support for A32 at any EL, x:[0:3] - All the levels below supplied ELx supports A32.

Type: `int8_t`

Default value: 3

**mdrar\_el1\_res0**

MDRAR\_EL1 is **RES0**.

Type: `bool`

Default value: `false`

**mdselr\_le\_16\_bps\_wps\_behaviour**

Behaviour of MDSELR\_EL1 and related traps/enables if fewer than 16 watchpoints and fewer than 16 breakpoints are implemented: 0 - MDSELR\_EL1 is stateful; 1 - MDSELR\_EL1, EBWE, FGTS are **RAZ/WI**, traps and enables do not apply; 2 - MDSELR\_EL1, EBWE, FGTS with checked traps.

Type: `uint8_t`

Default value: 0

**mec\_support\_level**

0 -> Memory Encryption Contexts not implemented, 1 -> LEGACY\_TZ\_EN mode i.e. MEC register fields are stateful but only supports secure/non-secure states, 2 -> Memory Encryption Contexts fully implemented (FEAT\_MEC).

Type: `uint8_t`

Default value: 0

**memory.acp.AxCACHE\_mask**

Used with `memory.acp.AxCACHE_pattern` to define which memory types the ACP port accepts. All transactions which do not satisfy  $(\text{AxCACHE} \& \text{mask}) == \text{pattern}$  will abort.

Type: `uint8_t`

Default value: 0

**memory.acp.AxCACHE\_pattern**

Used with `memory.acp.AxCACHE_mask` to define which memory types the ACP port accepts. All transactions which do not satisfy  $(\text{AxCACHE} \& \text{mask}) == \text{pattern}$  will abort.

Type: `uint8_t`

Default value: 0

**memory.l2\_cache.is\_inner\_cacheable**

L2 cache obeys inner cacheable attributes (rather than outer cacheable attributes).

Type: `bool`

Default value: `true`

**memory.l2\_cache.is\_inner\_shareable**

L2 cache obeys inner shareable attributes (rather than outer shareable attributes).

Type: `bool`

Default value: `true`

**memory\_tagging\_support\_level**

Specify the memory tagging extension support level: 0, not implemented. 1, instructions and registers only are implemented (FEAT\_MTE). 2, implemented (FEAT\_MTE2). 3, implemented with asymmetric handling of exceptions (FEAT\_MTE3). 4, implemented (FEAT\_MTE4).

Type: `uint8_t`

Default value: 0

**mixed\_endian**

Implement support for mixed endianness. 0, not supported. 1, supported at all exception levels. 2, supported at ELO only. Unsupported in the presence of Future Architecture Technologies (FAT).

Type: `uint8_t`

Default value: 1

**mops\_cpy\_block\_size**

Block size used for memcpy memory accesses.

Type: `uint8_t`

Default value: 64

**mops\_cpy\_default\_dir**

Default direction for non-overlapping memcpy operations: 0, forwards. 1, backwards.

Type: `uint8_t`

Default value: 0

**mops\_cpy\_handle\_async\_exceptions**

Handle any pending async exceptions after copying a block of data, instead of waiting until instruction end.

Type: `bool`

Default value: `false`

**mops\_cpy\_post\_size**

Percentage of data copied in memcpy 'E' instructions.

Type: `uint8_t`

Default value: 10

**mops\_cpy\_pre\_size**

Percentage of data copied in memcpy 'P' instructions.

Type: `uint8_t`

Default value: 10

**mops\_cpy\_pre\_size\_threshold**

Size threshold in Bytes for CPYP instructions.

Type: `uint32_t`

Default value: `0x0`

### **mops\_cpy\_single\_access**

Execute memcpy as a single read and single write access.

Type: `bool`

Default value: `false`

### **mops\_cpy\_write\_abort\_before\_read**

Report the data aborts and watchpoint of the write accesses, before those of the read accesses.

Type: `bool`

Default value: `false`

### **mops\_cpy\_zero\_size\_can\_fault**

Fault because of mismatched implementation option when the operation is of size 0.

Type: `bool`

Default value: `true`

### **mops\_exec\_order\_can\_fault**

Enable exception on the Main/Epilogue instruction when executed after a mismatched Prologue/Main in a CPY/SET sequence, or after another random instruction.

Type: `bool`

Default value: `false`

### **mops\_inst\_cpy\_zero\_size\_can\_fault**

Fault because of mismatched implementation option when `inst_cpy_size` is 0.

Type: `bool`

Default value: `true`

### **mops\_inst\_set\_zero\_size\_can\_fault**

Fault because of mismatched implementation option when `inst_set_size` is 0.

Type: `bool`

Default value: `true`

**mops\_mismatched\_page\_crossing\_access\_unpred**

Constrained unpredictable behaviour for FEAT\_MOPS instructions when crossing page boundary with different memory types, 0 : Memory block access uses the attributes of it's own address block  
1: Alignment Fault.

Type: `uint8_t`

Default value: 0

**mops\_mmu\_abort\_far\_aligned**

If true, in case of an MMU abort on a MOPS instruction, report FAR aligned to current translation granule.

Type: `bool`

Default value: `false`

**mops\_set\_block\_size**

Block size used for memset memory accesses.

Type: `uint8_t`

Default value: 64

**mops\_set\_handle\_async\_exceptions**

Handle any pending async exceptions after setting a block of data, instead of waiting until instruction end.

Type: `bool`

Default value: `false`

**mops\_set\_post\_size**

Percentage of data copied in memset 'E' instructions.

Type: `uint8_t`

Default value: 10

**mops\_set\_pre\_size**

Percentage of data copied in memset 'P' instructions.

Type: `uint8_t`

Default value: 10



**mops\_set\_single\_access**

Execute memset as a single read and single write access.

Type: `bool`

Default value: `false`

**mops\_set\_zero\_size\_can\_fault**

Fault because of mismatched implementation option when the operation is of size 0.

Type: `bool`

Default value: `true`

**mops\_setg\_unaligned\_does\_mismatch\_fault**

If true, in case of unaligned SETGM / SETGE, raise a mismatched memset exception because of impdef reasons, instead of alignment fault.

Type: `bool`

Default value: `false`

**mops\_wp\_far\_behaviour**

Set option for address stored in FAR/EDWARD after watchpoints hit by MOPS instructions

- FAR recorded matches lowest watchpointed address accessed by the instruction
- FAR recorded matches lowest address accessed by the instruction within same translation granule as watchpointed address
- FAR recorded matches highest watchpointed address accessed by the instruction that triggered the watchpoint.

Type: `uint8_t`

Default value: 0

**mpam\_at\_ops\_use\_target\_el**

If true, AT instruction table fetches use the owning translation regime's EL for MPAM PARTID/PMG selection.

Type: `bool`

Default value: `false`

**mpam\_bw\_bwa\_wd**

MPAM MPAMBWIDR\_EL1.BWA\_WD: The number of implemented bits in the bandwidth allocation fields {MPAMBWn\_ELx, MPAMBWSM\_EL1}.MAX and MPAMBWCAP\_EL2.CAP.

Type: `uint8_t`

Default value: 1

**mpam\_bw\_has\_hw\_scale**

MPAM Whether has hardware support for auto-scaling of {MPAMBWn\_ELx, MPAMBWSM\_EL1}.MAX and MPAMBWCAP\_EL2.CAP limits.

Type: `bool`

Default value: `false`

**mpam\_bw\_max\_lim**

MPAM the implemented maximum-bandwidth limit partitioning behaviors:- 0, Both soft limit and hard limit behaviors are implemented.- 1, Soft limit behavior is implemented.- 2, Hard limit behavior is implemented.

Type: `uint8_t`

Default value: 0

**mpam\_bw\_us\_frac**

MPAM MPAMBWIDR\_EL1.US\_FRAC: The fractional part of the window width in microseconds.

Type: `uint8_t`

Default value: 0

**mpam\_bw\_us\_int**

MPAM MPAMBWIDR\_EL1.US\_INT: The integer part of the window width in microseconds.

Type: `uint32_t`

Default value: 0x0

**mpam\_force\_ns\_rao**

Whether MPAM3\_EL3.FORCE\_NS bit is **RAO/WI**.

Type: `bool`

Default value: `false`

**mpam\_frac**

MPAM fractional revision number in ID\_AA64PFR1\_EL1.MPAM\_frac field. Combines with has\_mpam to give the mpam version mpam = false, mpam\_frac = 0 -> Not implemented mpam = false, mpam\_frac = 1 -> FEAT\_MPAMvOp1 mpam = true, mpam\_frac = 0 -> FEAT\_MPAMv1p0 mpam = true, mpam\_frac = 1 -> FEAT\_MPAMv1p1.

Type: `uint8_t`

Default value: 0

**mpam\_has\_altsp**

DEPRECATED: MPAMIDR\_EL1.HAS\_ALTSP is required when FEAT\_RME is implemented.

Type: `bool`

Default value: `false`

**mpam\_has\_bw\_ctrl**

MPAM Whether MPAMIDR\_EL1.HAS\_BW\_CTRL bit is set or clear.values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv9.3 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: `0`

**mpam\_has\_hcr**

MPAM Whether MPAMIDR\_EL1 HAS\_HCR bit is set or clear.

Type: `bool`

Default value: `false`

**mpam\_max\_partid**

MPAM Maximum PARTID Supported.

Type: `uint16_t`

Default value: `0xffff`

**mpam\_max\_pmg**

MPAM Maximum PMG Supported.

Type: `uint16_t`

Default value: `0xffff`

**mpam\_max\_vpmr**

MPAM Maximum VPMR Supported.

Type: `uint8_t`

Default value: `0`

**mpam\_truncate\_out\_of\_range\_virtid**

If true then truncates an out-of-range virtual identifier to least significant 12 bits. If false then out-of-range virtual identifier is replaced by the default virtual identifier.

Type: `bool`

Default value: `false`

### **`mpamidr_has_force_ns`**

Whether MPAMIDR\_EL1.HAS\_FORCE\_NS bit is set or clear.values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.6 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **`mpamidr_has_sdeflt`**

Whether MPAMIDR\_EL1.HAS\_SDEFLT bit is set or clear.values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.6 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **`mpamidr_has_tidr`**

Whether MPAMIDR\_EL1.HAS\_TIDR bit is set or clear.values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.6 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **`mpidr_layout`**

Layout of MPIDR. 0 AFF0 is CPUID, 1 AFF1 is CPUID.

Type: `uint8_t`

Default value: 0

### **`mpmm_accumulator_multiplier`**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator.value of n means the accumulator will use (n \* accumulator value) to calculate the mpmm threshold (MPMM).is provided as a fast model workaround to handle cases where execution of quantums in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

Type: `uint8_t`

Default value: 1

**mpmm\_config**

MPMMTUNE register value. The JSON schema is : . The value given for threshold value is just an indication, not specific to any core. This parameter is used only when has\_mpmm is set.

Type: string

Default value: N/A

**mte\_ctrl\_bits\_stateful\_level**

If memory\_tagging\_support\_level == 1, specify the MTE level that has control bits stateful in system registers.

Type: uint8\_t

Default value: 0

**mte\_report\_which\_failed\_address**

Set to <OPT>, <MOPS\_OPT>Applicable only for MTE synchronous check. OPT defines the range for the failing address to report and MOPS\_OPT defines the choice within that range for MOPS operations only.Non-MOPS operations report the first address in the range defined by OPT.OPT is set to "first" or "last". If "first" then report an address from the intersection of the first failed MTE granule and the transaction's range.If "last" then report an address from the intersection of the last failed MTE granule and the transaction's range.MOPS\_OPT is set to "mops\_first\_failing\_address\_in\_range" or "mops\_random\_address\_in\_range". If "mops\_first\_failing\_address\_in\_range" then report the first failing address in the range defined by OPT. If "mops\_random\_address\_in\_range" then report a random address within the range defined by OPT.

Type: string

Default value: "first, mops\_first\_failing\_address\_in\_range"

**mte\_tminline**

Value of CTR\_EL0.TminLine for reading purpose only. A value configured using this does not indicate the presence of separate tag cache. 0, TminLine evaluated from smallest data cache line.

Type: uint8\_t

Default value: 0

**mte\_unpred\_canonical\_s2\_unsupported**

When FEAT\_MTE\_CANONICAL\_TAGS is implemented, determines whether a region is canonically tagged (true) or untagged (false) when stage 1 is tagged but the combined attributes do not allow allocation tagging.

Type: bool

Default value: false

**mvbar\_reset\_is\_rvbar**

If true then the reset value of MVBAR is RVBAR, if false the reset value is **UNKNOWN**.

Type: bool

Default value: true

**non\_secure\_vgic\_alias\_when\_ns\_only**

If ! has\_el3 and only non-secure side exists, then the normal position of the VGIC is a secure alias. If this parameter is non-zero then in addition a non-secure alias of the VGIC will be placed at this position (aligned to 32 KB).

Type: uint64\_t

Default value: 0x0

**num\_acp**

Number of ACP ports.

Type: uint8\_t

Default value: 0

**num\_core\_reg\_slices**

Number of core register slices.

Type: uint8\_t

Default value: 0

**num\_l3\_slices**

Number of L3 cache slices.

Type: uint8\_t

Default value: 1

**num\_loregion\_descriptors**

Number of Limited Ordering Region descriptors implemented (if ARM v8.1 extensions are implemented) (FEAT\_LOR).

Type: uint8\_t

Default value: 0

**num\_loregions**

Number of Limited Ordering Regions implemented excluding background region (if ARM v8.1 extensions are implemented) (FEAT\_LOR).

Type: uint8\_t

Default value: 0

**num\_master**

Number of master interface.

Type: uint8\_t

Default value: 1

**num\_nodes**

Number of transport nodes. Zero implies direct-connect configuration.

Type: uint8\_t

Default value: 1

**num\_sf\_ways**

Number of snoop filter ways.

Type: uint8\_t

Default value: 4

**num\_trans\_reg\_slices\_horiz**

Number of transport register slices, horizontal.

Type: uint8\_t

Default value: 0

**num\_trans\_reg\_slices\_vert**

Number of transport register slices, vertical.

Type: uint8\_t

Default value: 0

**number\_of\_abl\_breakpoints**

if FEAT\_ABLE is implemented, Number of address matching breakpoints that support address linking.

Type: `uint8_t`

Default value: 0

### **number\_of\_error\_records**

Cores Number of Error records supported for RAS.

Type: `uint16_t`

Default value: 0x0

### **nv\_frac\_support\_level**

Support for a subset of FEAT\_NV and FEAT\_NV2 behaviours: 0 - Not implemented. 1 - Implemented. 2 - Implemented with FEAT\_NV2p1. 3 - Implemented with FEAT\_NV3, OPTIONAL from Armv9.6.

Type: `uint8_t`

Default value: 0

### **optimal\_alignment\_size**

Alignment boundary which does not incur additional performance penalty from ARMv8.5.- 1, architectural misalignment is used to set PMU event LDST\_ALIGN\_LAT and SPE event E[11]- 2, access crossing 4 byte boundary is used to set PMU event LDST\_ALIGN\_LAT and SPE event E[11]- 3, access crossing 8 byte boundary is used to set PMU event LDST\_ALIGN\_LAT and SPE event E[11].- 12, access crossing 4 Kbyte boundary is used to set PMU event LDST\_ALIGN\_LAT and SPE event E[11].

Type: `uint8_t`

Default value: 1

### **output\_attributes**

User-defined transform to be applied to bus attributes like ManagerID, ExtendedID or UserFlags. Currently, only works for MPAM Attributes encoding into bus attributes.

Type: `string`

Default value: "ExtendedID[62:55]=MPAM\_PMG, ExtendedID[54:39]=MPAM\_PARTID, ExtendedID[38]=MPAM\_NS"

### **pacm\_support\_level**

Implements PSTATE.PACM from ARMv9.5. 0: Not supported, 1: Trivial implementation when FEAT\_PAuth\_LR and FEAT\_PACIMP are supported, 2: Full implementation when FEAT\_PAuth\_LR is supported.

Type: `uint8_t`



Default value: 2

### **page\_based\_hardware\_attributes**

Implement the page based hardware attributes from ARMv8.2. This parameter indicates which page table bits are available for hardware, where bits[3:0] correspond to PTE[62:59] and to TCR\_ELx.HWUnyy (FEAT\_HPDS2).

Type: `uint8_t`

Default value: 0

### **pan\_applies\_before\_oa\_space\_checks**

Apply effects of PAN on S1 indirect permissions before OA space security checks.

Type: `bool`

Default value: `true`

### **pan\_removes\_priv\_rw\_if\_unpriv\_resvd\_value**

For regimes EL1&0/EL2&0, PSTATE.PAN = 1, whether PrivRW is removed when S1UnprivBasePerm is reserved.

Type: `bool`

Default value: `true`

### **pan\_stage1\_in\_realm\_el2\_0\_is\_uxn**

If FEAT\_PAN3 is implemented, whether stage1 translation in the Realm EL2&0 regime that resolves to a NS address is treated as unprivileged execute-never.

Type: `bool`

Default value: `false`

### **par\_ns\_set\_unknown\_bit**

Whether NS bit of PAR is set/clear when executing AT to perform non-secure regime translation. When true, NS is set to 1 else 0.

Type: `bool`

Default value: `true`

### **par\_nse\_set\_unknown\_bit**

Whether NSE bit of PAR is set/clear when executing AT operation on secure, non-secure or realm translation regime. When true, NSE is set to 1 else 0.

Type: `bool`

Default value: `false`

**pchannel\_treat\_simreset\_as\_poreset**

Register core as ON state to cluster with simulation reset.

Type: `bool`

Default value: `false`

**periph\_address\_end**

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

Type: `uint64_t`

Default value: `0x0`

**periph\_address\_start**

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

Type: `uint64_t`

Default value: `0x0`

**periph\_width\_in\_bits**

Peripheral port width in bits. 0 when no peripheral port is configured. 64 and 256 are allowed when peripheral port is configured.

Type: `uint16_t`

Default value: `0`

**permission\_overlay\_s1\_support\_level**

Support for Stage 1 Permission Overlay: 0 - None. 1 - Support FEAT\_S1POE. 2 - Support FEAT\_S1POE2.

Type: `uint8_t`

Default value: `0`

**pfar\_is\_valid**

**IMPLEMENTATION DEFINED** choice to configure ESR\_ELx.PFV: whether PFAR\_ELx is valid or **UNKNOWN** when ESR\_ELx.PFV is not forced to be 0.

Type: `bool`

Default value: `true`

**pfr1\_csv2\_frac**

Fractional revision number ID\_AA64PFR1\_EL1.CSV2\_frac when ID\_AA64PFR0\_EL1.CSV==1 for CSV2 extension (FEAT\_CSV2\_1p1, FEAT\_CSV2\_1p2).

Type: `uint8_t`

Default value: 0

**plb\_dpot\_enabled**

Enable PLB for POE2 DPOT lookups. NOTE: setting this parameter to false will significantly degrade model performance.

Type: `bool`

Default value: `true`

**plb\_irt\_enabled**

Enable PLB for POE2 IRT lookups. NOTE: setting this parameter to false will significantly degrade model performance.

Type: `bool`

Default value: `true`

**plb\_ttt\_enabled**

Enable PLB for POE2 TTT lookups. NOTE: setting this parameter to false will significantly degrade model performance.

Type: `bool`

Default value: `true`

**plbi\_invalid\_xt**

If true, PLBI instructions with invalid Xt are treated as UNDEF.

Type: `bool`

Default value: `false`

**pmb\_idr\_external\_abort**

Describes how the PE manages External aborts on writes made by the Statistical Profiling Extension to the Profiling Buffer. 0, External abort is reported to SPE, From Armv8.8 and Armv9.3, the value 0 is not permitted. 1, External abort is ignored. 2, The External abort generates an SError and the error is not reported to SPE.

Type: `uint8_t`

Default value: 0

**pmb\_idr\_flag\_updates**

Defines whether the address translation performed by the Profiling Buffer manages the Access Flag and dirty state.

Type: `bool`

Default value: `true`

**pmbsr\_dl\_razwi**

Whether PMBSR\_ELx.DL is **RAZ/WI** or behaves as specified, indicating partial loss of a record due to a buffer management event or external abort.

Type: `bool`

Default value: `false`

**pmbsr\_ea\_razwi**

Whether PMBSR\_ELx.EA is **RAZ/WI** or set as the result of an external abort.

Type: `bool`

Default value: `false`

**pmcr\_disable\_events\_export**

If true, export for PMU events is disabled. This configures PMCFGR.EX field.

Type: `bool`

Default value: `true`

**pmmir\_el1\_bus\_slots**

Largest value by which BUS\_ACCESS can increment over BUS\_CYCLES cycles. From v8.7 PMU extension.

Type: `uint16_t`

Default value: `0`

**pmmir\_el1\_bus\_width**

Width, in bytes, of accesses counted by BUS\_ACCESS. From v8.7 PMU extension.

Type: `uint16_t`

Default value: `0x0`

**pms\_idr\_max\_size**

Defines largest size for a single SPE record (rounded up to a power of 2).

Type: `uint8_t`

Default value: 6

### **`pmu-num_counters`**

Number of PMU counters implemented.

Type: `uint8_t`

Default value: 8

### **`pmu_async_exception_delay`**

Configure PMU asynchronous exception delay in CPU cycles (FEAT\_SEBEP).

Type: `uint32_t`

Default value: 0x0

### **`pmu_cycle_counter_counts_actual_cycles`**

If true and Timing annotation is enabled, PMU cycle counter counts actual cycles, otherwise counts instructions executed.

Type: `bool`

Default value: `false`

### **`pmu_has_chain_event`**

PMU (if present) implements event number 0x1e, CHAIN.

Type: `bool`

Default value: `true`

### **`pmu_precise_events`**

"Configure v9.4 Precise PMU events. {"pmu\_events":["SW\_INCR", "PC\_WRITE\_RETIRED", "BR\_RETIRED", "BR\_IND\_RETIRED", "BR\_RETURN\_RETIRED", "BR\_RETURN\_ANY\_RETIRED", "BR\_IND\_TAKEN\_RETIRED", "LD\_RETIRED", "ST\_RETIRED", "UNALIGNED\_LD\_ST", "INST\_RETIRED", "EXCEP\_TAKEN", "EXCEP\_RETURN", "CHAIN"]}".

Type: `string`

Default value: N/A

### **`pmu_threshold_bit_width`**

Implement FEAT\_PMUv3\_TH and if so the width of PMEVTYPER<n>\_ELO.TH in bits. 0, not implemented. 1-12 number of bits in PMEVTYPER<n>\_ELO.TH.

Type: `uint8_t`

Default value: 0

**poe2\_mmu\_fault\_far\_aligned**

If true, in case of an MMU fault on an access to IRT, DPOT or TTT, report FAR aligned to current translation granule or POE2 table size, whichever is smaller. By default, this is set to false - with FAR reporting the exact faulting VA.

Type: `bool`

Default value: `false`

**poison\_range\_end\_addr**

End PA of poisoned range.

Type: `uint64_t`

Default value: `0x0`

**poison\_range\_start\_addr**

Start PA of poisoned range.

Type: `uint64_t`

Default value: `0x0`

**preserve\_cache\_contents\_over\_warm\_reset**

Preserve cache contents over warm reset by ignoring the value of `ram_clear_on_reset_disable` signal in the cache.

Type: `bool`

Default value: `false`

**pseudo\_fault\_generation\_feature\_register**

ARMv8.4 Standard Pseudo-fault generation feature register values. JSON schema for the parameter value is: `[{"OF":false, "UC":false, "UEU":false, "UER":false, "UEO":false, "DE":false, "CE":0x0, "CI":false, "ER":false, "PN":false, "AV":false, "MV":false, "SYN":false, "R":false, "NA":false}, other_pseudo-fault_generating_features_register_values]`. Where OF, UC, UEU, UER, UEO, DE, CI, ER, PN, AV, MV, SYN, and R have valid `false`(NOT\_SUPPORTED) and `true`(FEATURE\_CONTROLLABLE), where CE can have `0`(NOT\_SUPPORTED), `1`(NONSPECIFIC\_CE\_SUPPORTED) and `3`(TRANSIENT\_OR\_PERSISTENT\_CE\_SUPPORTED) and NA can have `false`(component fakes detection on next access) or `true`(component fakes detection spontaneously). Effective only when ERXFR's INJ field allows it or `has_ras_fault_injection` is true.

Type: `string`

Default value: `N/A`

**pstate\_btype\_on\_illegal\_eret**

DEPRECATED. Please use `pstate_unknown_fields_on_illegal_eret` instead. This parameter will be ignored if both are used. In case of an illegal eret, what value to use to update PSTATE.BTYPE. 0: Set to 0, 1: Update with SPSR\_ELx.BTYPE, 2: Unchanged.

Type: `uint8_t`

Default value: 0

**pstate\_pm\_reset**

Reset value of PSTATE.PM.

Type: `bool`

Default value: `false`

**pstate\_ssbs\_reset**

Reset value of `pstate.ssbs`.

Type: `bool`

Default value: `false`

**pstate\_ssbs\_type**

Implement speculative store bypass safe feature from ARMv8.5. 0, Not supported. 1, Supported without MSR/MRS access to SSBS (FEAT\_SSBS). 2, fully supported (FEAT\_SSBS2).

Type: `uint8_t`

Default value: 0

**pstate\_unknown\_fields\_on\_illegal\_eret**

For an illegal eret, each unknown field in PSTATE can be set to "set\_to\_0", "set\_to\_spsr\_elx", or "unchanged". If a field is not specified but its relevant feature is enabled, it will default to "set\_to\_spsr\_elx". Fields for disabled features are ignored. Example: {"BTYPE": "set\_to\_0", "DIT": "set\_to\_spsr\_elx", "PACM": "set\_to\_0", "SSBS": "set\_to\_spsr\_elx", "TCO": "unchanged", "UAO": "set\_to\_spsr\_elx"}.

Type: `string`

Default value: N/A

**ptw\_latency**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

**randomize\_unknowns\_at\_reset**

Will fill in unknown bits in registers at reset with random value using register\_reset\_data as seed, it overrides scramble\_unknowns\_at\_reset.

Type: `bool`

Default value: `false`

**ras\_aderr\_anerr\_controls\_are\_same**

If true and FEAT\_ADERR and FEAT\_ANERR is implemented then ADERR and ANERR controls should always be set to the same value (FEAT\_ADERR) (FEAT\_ANERR).

Type: `bool`

Default value: `false`

**ras\_err\_registers\_undef\_if\_no\_error\_records**

If true, all RAS error record registers, along with ERRSELR\_EL1, will be undefined if ERRIDR\_EL1 indicates that zero error records are implemented.

Type: `bool`

Default value: `false`

**ras\_errselr\_undef\_if\_no\_error\_records**

If true, ERRSELR\_EL1 will be undefined if ERRIDR\_EL1 indicates that zero error records are implemented.

Type: `bool`

Default value: `false`

**ras\_extra\_configurations**

Miscellaneous configurations for error records. An array of JSON objects. Note for ERXCTLR\_EL1 register it only allows to define the mask value for the IMPDEF fields, ie bits [63:32] and bit 1, but its reset value applies on all fields. Note for ERXMISCN masks - these are 64 bit masks covering the 64 bit registers ERXMISCN\_EL1. E.g. [{"Index": 0, "ERXMISCO\_mask": 0x0, "ERXMISCO\_reset": 0x0, "ERXMISC1\_mask": 0x0, "ERXMISC1\_reset": 0x0, "ERXMISC2\_mask": 0x0, "ERXMISC2\_reset": 0x0, "ERXMISC3\_mask": 0x0, "ERXMISC3\_reset": 0x0, "ERXCTLR\_EL1\_mask": 0x0, "ERXCTLR\_EL1\_reset": 0x0}, {"Index": 1, "ERXMISCO\_mask": 0x0, "ERXMISCO\_reset": 0x0, "ERXSTATUS\_IERR\_mask": 0x300}].

Type: `string`

Default value: `N/A`



**ras\_frac**

0, No additional feature implemented. 1, Additional ERXMISC, ERXPFG registers and FaultInjection trap from RAS v1.1. implemented.

Type: `uint8_t`

Default value: 0

**ras\_log2\_fault\_granule\_size**

Log2 of the RAS fault granule size in Bytes.

Type: `uint8_t`

Default value: 12

**ras\_mmap\_address**

Base address of memory mapped RAS Registers for each core on system bus. 0 means the RAS is not mapped, otherwise the address must be at least 4KB aligned or more depending upon the features implemented. JSON schema for the parameter value is: `{"format": "all_addrs_are_absolute_wrt_systembus", "cores": [{"ras": 0x0}, {"ras": 0x0}, {"ras": 0x0}, {"ras": 0x0}]}`.

Type: `string`

Default value: N/A

**ras\_pfg\_clock\_mhz**

RAS Pseudo-Fault generation clock rate in MHz.

Type: `uint8_t`

Default value: 24

**ras\_pfg\_err\_opt**

When accessing RAS Pseudo-fault Generation register, one of the following errors occurs under a certain condition: 0 - **NOP** Error, 1 - **UNDEF** Error, 2 - **OK**.

Type: `uint8_t`

Default value: 1

**ras\_report\_aligned\_pa\_in\_pfar**

If true, the PFAR\_ELx register reports the PA aligned to the RAS fault granule size on a sync external abort or SError exception.

Type: `bool`

Default value: `false`

**register\_reset\_data**

Data used to fill register bits when they become **UNKNOWN** at reset.

Type: `uint64_t`

Default value: `0x0`

**register\_reset\_data\_hi**

Data used to fill the upper-half of 128-bit registers when the bits become **UNKNOWN** at reset.

Type: `uint64_t`

Default value: `0x0`

**report\_iside\_cmo\_ifsr**

fault info for an iside cache maintenance operation is reported in the IFSR.

Type: `bool`

Default value: `true`

**report\_second\_access\_align\_fault\_non\_atomic\_pair\_access**

If true, the non-atomic load/store pair accesses report the 2nd register as faulting address for an alignment fault. This is IMP-DEF behavior as defined in FEAT\_LRCPC3.

Type: `bool`

Default value: `false`

**report\_second\_access\_mmu\_fault\_non\_atomic\_pair\_access**

If true, the non-atomic load/store pair accesses report the 2nd register as faulting address for an MMU fault. This is IMP-DEF behavior as defined in FEAT\_LRCPC3.

Type: `bool`

Default value: `false`

**reported\_fp\_revision**

Purely cosmetic patch level value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored. Updates the FPSID.revision value.

Type: `int8_t`

Default value: `-1`

**reported\_patch\_level**

Purely cosmetic patch level value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: `int8_t`

Default value: -1

**reported\_revision\_number**

Purely cosmetic revision number value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: `int8_t`

Default value: -1

**reserved\_HMC\_SSC\_PAC\_treated\_disabled**

When `DBG[B|W]CR.{HMC,SSC,PAC}` bits configuration is reserved, this parameter controls whether breakpoints/watchpoints are treated as Disabled or not.

Type: `bool`

Default value: `false`

**restore\_fpsr\_on\_trapped\_fp\_exception**

If true, FPSR is restored to the value of the FPSR immediately before the instruction that generated the trapped floating-point exception.

Type: `bool`

Default value: `false`

**restriction\_on\_speculative\_execution**

Implements the ARMv8.5 security feature (Restrictions on the effects of speculation), `ID_AA64PFR0_EL1.CSV2`: 0: No disclosure whether branch targets trained in one context can affect speculative execution in a different context, 1: Branch targets trained in one context cannot affect speculative execution in a different hardware described context (`SCXTNUM_ELx` not supported), 2: Branch targets trained in one context cannot affect speculative execution in a different hardware described context (`SCXTNUM_ELx` supported) (`FEAT_CSV2`, `FEAT_CSV2_2`), 3: `FEAT_CSV2_3` is supported.

Type: `uint8_t`

Default value: 0

**restriction\_on\_speculative\_execution\_aarch32**

Implements the ARMv8.5 security feature (Restrictions on the effects of speculation), ID\_PFR0.CSV2: 0: No disclosure whether branch targets trained in one context can affect speculative execution in a different context, 1: Branch targets trained in one context cannot affect speculative execution in a different hardware described context, 2: Branch targets trained in one context cannot affect speculative execution in a different hardware described context or at a different address in the same hardware described context (FEAT\_CSV2, FEAT\_CSV2\_2).

Type: `uint8_t`

Default value: 0

**revision\_number**

Revision number of TRM implemented by the model. Changing the value can change the model behaviour. Visible as the Variant field in MIDR/MIDR\_EL1. Corresponds to the revision number X in rXpY.

Type: `uint8_t`

Default value: 0

**rgsr\_res0\_stateful**

Mask of RGSR **RES0** bits read of which return last written value.

Type: `uint64_t`

Default value: 0x0

**rme\_ctrl\_bits\_stateful\_level**

If `rme_support_level == 1`, specify the RME level that has control bits stateful in system registers. 0 - all until v9.2, 1 - v9.4 including FEAT\_RME\_GDI bits in MFAR\_EL3 and PFAR\_EL2.

Type: `uint8_t`

Default value: 0

**rme\_default\_mecid\_nonsecure**

Default MECID value for NON-SECURE PAS.

Type: `uint32_t`

Default value: 0x0

**rme\_default\_mecid\_realm**

Default MECID value for REALM PAS.

Type: `uint32_t`

Default value: 0x0

**rme\_default\_mecid\_root**

Default MECID value for ROOT PAS.

Type: uint32\_t

Default value: 0x0

**rme\_default\_mecid\_secure**

Default MECID value for SECURE PAS.

Type: uint32\_t

Default value: 0x0

**rme\_full\_is\_tagged\_nsh\_treated\_as\_tagged**

Whether a tagged NonShared memory attribute is treated as tagged or not. Does nothing if effective RME support is not full.

Type: bool

Default value: false

**rme\_level0\_gpt\_size**

The range of address space protected by each entry in the level 0 GPT (0->1GB 1->16GB, 2->64GB, 3->512GB).

Type: uint8\_t

Default value: 0

**rme\_mecid\_width**

Width of MECID in bits.

Type: uint8\_t

Default value: 1

**rme\_nsh\_cacheable\_is\_shareable**

If true, NSH cacheable becomes shareable cacheable (FEAT\_RME).

Type: bool

Default value: false

**rme\_support\_level**

0 -> Realm management extension not implemented, 1 -> LEGACY\_TZ\_EN mode i.e. RME register fields are stateful but only supports secure/non-secure states, 2 -> Realm management extension fully implemented (FEAT\_RME).

Type: `uint8_t`

Default value: 0

**rnr\_always\_implemented**

Always implement RMR\_ELx, RMR, or HRMR at the highest implemented exception level, even if that exception level cannot use both AArch32 and AArch64.

Type: `bool`

Default value: `false`

**rndr\_rndrrs\_seed**

Initial seed for random engine used in RNDR register.

Type: `uint64_t`

Default value: 0x0

**s1\_align\_memtype\_fault\_prio\_more\_than\_s2\_perm\_fault\_on\_s1\_walk**

If true, s1 alignment fault has priority over s2 permission faults.

Type: `bool`

Default value: `true`

**s1\_perm\_fault\_prio\_more\_than\_s2\_perm\_fault\_on\_s1\_walk**

If true, s1 permission fault has priority over s2 on s1 translation table walk permission faults.

Type: `bool`

Default value: `false`

**s1\_unsupported\_atomic\_fault\_for\_ls64\_prio\_more\_than\_s2\_perm\_fault**

If true, unsupported atomic/exclusive faults due to LS64 instructions at Stage 1 have higher priority than permission fault at Stage 2.

Type: `bool`

Default value: `false`

**scheduler\_mode**

Control the interleaving of instructions in this processor. 0, default long quantum. 1, low latency mode, short quantum and signal checking. 2, lock-breaking mode, long quantum with additional context switches near load-exclusive instructions. WARNING: This parameter is intended for validation purposes and may result in unwanted behaviour if altered!.

Type: `uint8_t`

Default value: 0

**scr\_nET\_writeable**

Whether SCR.nET is writeable. Writing to it is purely cosmetic (nET behavior not implemented).

Type: `bool`

Default value: `false`

**scramble\_unknowns\_at\_reset**

Will fill in unknown bits in registers at reset with `register_reset_data`.

Type: `bool`

Default value: `true`

**scu\_cache\_protection**

SCU-L3 is configured with ECC if true.

Type: `bool`

Default value: `false`

**serror\_clear\_delay**

Delay for clearing of SError if SError is level-triggered, in cpu cycles.

Type: `uint32_t`

Default value: 0x0

**set\_mops\_option**

Set option for Armv8.8 SET(FEAT\_MOPS). 0, use default(i.e. use value configured through `has_mops_option`). 1, implemented using Option A. 2, implemented using Option B.

Type: `uint8_t`

Default value: 0

**set\_rasv10\_for\_armv84\_and\_higher**

ARMv8.4 mandates RAS System Architecture v1.1, but when there are no error records and FEAT\_DoubleFault is not implemented then there is no functional difference between the RAS System Architecture v1.0 (that is, the RAS extension as in pre-ARMv8.4 implementations) and the RAS System Architecture v1.1 (also known as FEAT\_RASv1p1). This flag if true will set the RAS ID to declare RAS v1.0 rather than RAS v1.1 for ARMv8.4 and higher implementations. If this is set and the core does not conform to the restrictions then this parameter is ignored.

Type: `bool`

Default value: `false`

**setg\_mops\_option**

Set option for Armv8.8 SETG(FEAT\_MOPS). 0, use default(i.e. use value configured through `has_mops_option`). 1, implemented using Option A. 2, implemented using Option B.

Type: `uint8_t`

Default value: 0

**skip\_trace\_on\_write\_to\_osecctr\_el1\_when\_oslock\_is\_unlocked**

If `OSLSR_EL1.OSLK == 0`, then `OSECCR_EL1` returns an unknown value on reads and ignores writes. When true, also skips the traces on writes to `OSECCR_EL1` when `OSLSR_EL1.OSLK == 0`.

Type: `bool`

Default value: `false`

**spe\_counter\_size**

Size of counter packet payload in Statistical Profiling Extension- 1, Counter packet payloads are 12-bit saturating counters- 2, Counter packet payloads are 16-bit saturating counters.

Type: `uint8_t`

Default value: 1

**spmu\_support\_level**

Implement System PMU: 0: Not supported, 1: v8.9 System PMU Extension is implemented (FEAT\_SPMU), 2: v9.5 System PMU2 Extension is implemented (FEAT\_SPMU2) values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.9 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0



**spsr\_el3\_is\_mapped\_to\_spsr\_mon**

Whether SPSR\_EL3 is mapped to AArch32 register SPSR\_mon.

Type: `bool`

Default value: `true`

**spsr\_m4\_res0**

Whether SPSR\_ELx.M[4] bit should be **RES0** for AARCH64 only implementations.

Type: `bool`

Default value: `false`

**squash\_iside\_innerWTouterWT\_to\_non\_cacheable**

Whether inner WriteThrough outer WriteThrough from iside is treated NonCacheable in L2,L3.

Type: `bool`

Default value: `false`

**stage12\_tlb\_size**

Number of stage1+2 tlb entries. Valid values are 0 or  $\geq 4$ .

Type: `uint32_t`

Default value: `0x4000`

**stage1\_tlb\_size**

Number of stage1 only tlb entries. Valid values are 0 or  $\geq 4$ .

Type: `uint32_t`

Default value: `0x0`

**stage1\_walkcache\_size**

Number of stage1 only walk cache entries.

Type: `uint32_t`

Default value: `0x0`

**stage2\_tlb\_size**

Number of stage2 only tlb entries.

Type: `uint32_t`

Default value: `0x0`

**stage2\_walkcache\_size**

Number of stage2 only walk cache entries.

Type: `uint32_t`

Default value: `0x0`

**statistical\_profiling\_buffer\_alignment**

Statistical profiling alignment constraint for sample buffer.

Type: `uint16_t`

Default value: `0x1`

**statistical\_profiling\_datasrc\_payload\_size**

Statistical profiling size in bytes of Data Source packet' payloads.

Type: `uint8_t`

Default value: `1`

**statistical\_profiling\_random\_interval\_is\_separate**

Statistical profiling random interval gets added to the main timer interval(false) or (true) runs as separate timer.

Type: `bool`

Default value: `false`

**statistical\_profiling\_recommended\_min\_sampling**

Statistical profiling recommended minimum sampling interval.

Type: `uint16_t`

Default value: `0x100`

**stex\_fail\_suppress\_sync\_data\_aborts**

If true, synchronous data aborts are not reported if store exclusive fails.

Type: `bool`

Default value: `false`

**store\_excl\_fail\_tag\_check\_action**

Behavior of tag check by core when a store exclusive fails. 0, Tag check ignored, 1, Tag check done if exclusive fails by global monitor.

Type: `uint8_t`

Default value: 0

### **`strex_fail_can_hit_watchpoint`**

If true, a strex fail can hit watchpoint.

Type: `bool`

Default value: `false`

### **`stzgm_reports_fault_address_from_reg_arg`**

Which faulting address should be reported in FAR\_ELx on a failed STZGM: 0: the lowest aligned addr to DCZID-log2-block-size, 1: the addr held in the register argument, 2: if it is a tag-check fault, the addr aligned to DCZID-log2-block-size, otherwise the addr held in the register argument.

Type: `uint8_t`

Default value: 0

### **`supports_multi_threading`**

Sets the MPIDR.MT bit. Setting this to true hints the the cluster is multi-threading compatible.

Type: `bool`

Default value: `false`

### **`sve.clear_constrained_lanes`**

When a constrained vector length increases, previously inaccessible bits are set to zero. Possible values are: 0=never, 1=always, 2=if the register was written to while the vector length was constrained.

Type: `uint8_t`

Default value: 0

### **`sve.combine_movprfx_and_destructive`**

Attempt to combine the execution of MOVPRFX and the destructively-encoded instruction that follows it.

Type: `bool`

Default value: `false`

### **`sve.disable_speculative_accesses`**

All speculative memory accesses behave as though faulting, without accessing memory.

Type: `bool`

Default value: `false`

### **`sve.enable_at_reset`**

Start with system registers set up for Scalable Vector Extension use.

Type: `bool`

Default value: `false`

### **`sve.ffr_16b_pattern_UNKNOWN`**

A specific 16-bit **UNKNOWN** value that is used by parameter `force_UNKNOWN_to_ffr`.

Type: `uint16_t`

Default value: `0x0`

### **`sve.force_UNKNOWN_to_ffr`**

Governs behavior if WRFFR writes a non-monotonic value to FFR. Possible values are: 0 - Write non-canonical value to FFR, 1 - Overwrite FFR with a specific pattern of 16-bit **UNKNOWN** value. See `ffr_16b_pattern_UNKNOWN`, 2 - Clear all bits above first zero 3 - Set all bits after first one.

Type: `uint8_t`

Default value: `0`

### **`sve.fp_exception_report_lowest`**

If true, for multiple trapped FP exceptions, report the lowest lane in VECITR. Otherwise, report the highest.

Type: `bool`

Default value: `false`

### **`sve.fp_exception_set_tfv`**

Set `ESR_ELx.TFV` during FP exception. Trapped exception flags are valid.

Type: `bool`

Default value: `true`

### **`sve.fp_exception_set_vecitr`**

If true, set `ESR_ELx.VECITR` during FP exception. Otherwise, set **RES0**.

Type: `bool`

Default value: `false`

**sve.has\_b16b16**

Whether FEAT\_SVE\_B16B16 is implemented. Possible values are: 0 - Not implemented, 1 - Implemented if either FEAT\_SVE2 or FEAT\_SME2 is implemented.

Type: `uint8_t`

Default value: 0

**sve.has\_b16mm**

Whether FEAT\_SVE\_B16MM is implemented. Possible values are: 0 - Not implemented, 1 - Implemented if either FEAT\_SVE2 or FEAT\_SME2 is implemented.

Type: `uint8_t`

Default value: 0

**sve.has\_bfscale**

Whether FEAT\_SVE\_BFSCALE is implemented. Possible values are: 0 - Not implemented, 1 - Implemented if either FEAT\_SVE2 or FEAT\_SME2 is implemented.

Type: `uint8_t`

Default value: 0

**sve.has\_sme**

Whether SME is implemented (FEAT\_SME).

Type: `bool`

Default value: `false`

**sve.has\_sme2**

Whether SME2 is implemented (FEAT\_SME2).

Type: `bool`

Default value: `false`

**sve.has\_sme\_b16b16**

Whether FEAT\_SME\_B16B16 is implemented. Possible values are: 0 - Not implemented, 1 - Implemented if FEAT\_SME2 is implemented.

Type: `uint8_t`

Default value: 0

**sve.has\_sme\_f16f16**

Whether FEAT\_SME\_F16F16 is implemented. Possible values are: 0 - Not implemented, 1 - Implemented if FEAT\_SME2 is implemented.

Type: `uint8_t`

Default value: 0

**sve.has\_sme\_f64f64**

If SME is implemented, whether double-precision FMOPA and FMOPS are implemented.

Type: `uint8_t`

Default value: 1

**sve.has\_sme\_f8f16**

If SME2 is implemented, whether FEAT\_SME\_F8F16 is implemented.

Type: `uint8_t`

Default value: 1

**sve.has\_sme\_f8f32**

If SME2 is implemented, whether FEAT\_SME\_F8F32 is implemented.

Type: `uint8_t`

Default value: 1

**sve.has\_sme\_fa64**

Whether FEAT\_SME\_FA64 is implemented.

Type: `bool`

Default value: `false`

**sve.has\_sme\_i16i64**

If SME is implemented, whether instructions that accumulate 16-bit integer outer products into 64-bit integer tiles are implemented.

Type: `uint8_t`

Default value: 1

**sve.has\_sme\_lutv2**

Whether FEAT\_SME\_LUTv2 is implemented.

Type: `bool`

Default value: `false`

#### **`sve.has_sme_mop4`**

Whether FEAT\_SME\_MOP4 is implemented. Possible values are 0 - Implemented if FEAT\_SME2p2 is implemented, 1 - Implemented if FEAT\_SME2p1 is implemented.

Type: `uint8_t`

Default value: 0

#### **`sve.has_sme_priority_control`**

Whether SME Priority Control is implemented.

Type: `bool`

Default value: `true`

#### **`sve.has_sme_tmop`**

Whether FEAT\_SME\_TMOP is implemented. Possible values are 0 - Implemented if FEAT\_SME2p2 is implemented, 1 - Implemented if FEAT\_SME2p1 is implemented.

Type: `uint8_t`

Default value: 0

#### **`sve.has_ssve_aes`**

Indicates support for SVE2 and SME2 AES instructions when the PE is in Streaming SVE mode (FEAT\_SSVE\_AES).

Type: `uint8_t`

Default value: 0

#### **`sve.has_ssve_bit_perm`**

Whether FEAT\_SSVE\_BitPerm is implemented. Possible values are 0 - Not Implemented, 1 - Implemented if FEAT\_SME2p1 is implemented.

Type: `uint8_t`

Default value: 0

#### **`sve.has_ssve_fexpa`**

Whether FEAT\_SSVE\_FEXPA is implemented. Possible values are 0 - Implemented if FEAT\_SME2p2 is implemented, 1 - Implemented if FEAT\_SME2p1 is implemented.

Type: `uint8_t`

Default value: 0

### **sve.has\_sve2**

Whether SVE2 is implemented (FEAT\_SVE2).

Type: `bool`

Default value: `false`

### **sve.has\_sve2\_aes**

If SVE2 is implemented, whether AES instructions are implemented. Possible values are: 0 - not implemented, 1 - SVE2 AESE, AESD, AESMC, and AESIMC are implemented (FEAT\_SVE\_AES), 2 - Same as 1 but in addition SVE2 PMULLB and PMULLT with 64-bit source are implemented, 3 - Same as 2 but SVE2 64-bit source element PMLALB and PMLALT instruction variants are implemented (FEAT\_SVE\_PMULL128).

Type: `uint8_t`

Default value: 2

### **sve.has\_sve2\_bit\_perm**

If SVE2 is implemented, whether BitPerm instructions are implemented (FEAT\_SVE\_BitPerm).

Type: `bool`

Default value: `true`

### **sve.has\_sve2\_sha3**

If SVE2 is implemented, whether SHA3 instructions are implemented (FEAT\_SVE\_SHA3).

Type: `bool`

Default value: `true`

### **sve.has\_sve2\_sm4**

If SVE2 is implemented, whether SM4 instructions are implemented (FEAT\_SVE\_SM4).

Type: `bool`

Default value: `true`

### **sve.has\_sve\_bf16**

Whether SVE BFloat16 instructions are implemented.

Type: `bool`

Default value: `true`



**sve.has\_sve\_extended\_bf16**

Deprecated: to enable FEAT\_EBF16, use CPU parameter has\_ebf16. Whether Extended BFloat16 instructions are implemented. Possible values are: 0 - Disabled, 1 - Enabled if SME or SVE is implemented, 2 - Enabled if SME is implemented.

Type: `uint8_t`

Default value: 2

**sve.has\_sve\_f16f32mm**

Whether the SVE half-precision to single-precision Matrix Multiply instructions are implemented (FEAT\_F16F32MM).

Type: `bool`

Default value: `false`

**sve.has\_sve\_mm\_f32**

Whether the SVE FP32 Matrix Multiply instructions are implemented (FEAT\_F32MM).

Type: `bool`

Default value: `true`

**sve.has\_sve\_mm\_f64**

Whether the SVE FP64 Matrix Multiply instructions are implemented (FEAT\_F64MM).

Type: `bool`

Default value: `true`

**sve.has\_sve\_mm\_i8**

Whether the SVE Int8 Matrix Multiply instructions are implemented (FEAT\_I8MM).

Type: `bool`

Default value: `true`

**sve.movprfx\_unpredictable\_behavior**

Defines the behavior of MOVPRFX and the instruction it immediately precedes when the behavior is **CONSTRAINED UNPREDICTABLE**. Possible values are: 0 - UNDEF execution from MOVPRFX, 1 - MOVPRFX and second half of instruction executes as **NOP**, 2 - **NOP** MOVPRFX only, 3 - UNDEF execution from MOVPRFX unless otherwise trapped.

Type: `uint8_t`

Default value: 0

**sve.predicated\_sp\_align\_check\_behaviour**

Governs behavior of SP alignment checking for predicated memory accesses. Possible values are: 0 - Always perform, 1 - Skip if governing predicate is 0, 2 - Skip for contiguous accesses if governing predicate is 0, 3 - Skip for gather/scatter accesses if governing predicate is 0.

Type: `uint8_t`

Default value: 0

**sve.relax\_sme\_watchpoint\_matching\_16**

Whether memory accesses through Z and P registers in Streaming Mode and all accesses through ZA match watchpoints rounded to 16-byte alignment.

Type: `bool`

Default value: `false`

**sve.relax\_sve\_watchpoint\_matching\_16**

If FEAT\_DEBUGv8p9 is implemented, whether memory accesses through Z and P registers outside Streaming Mode match watchpoints rounded to 16-byte alignment.

Type: `bool`

Default value: `false`

**sve.sm\_tag\_checked**

Whether SME, SVE, and SIMD&FP load and store instructions executed when the PE is in Streaming SVE mode perform a Tag Check.

Type: `bool`

Default value: `true`

**sve.sme2\_version**

The version of SME2 if implemented. Possible values are: 0 - FEAT\_SME2, 1 - FEAT\_SME2p1, 2 - FEAT\_SME2p2, 3 - FEAT\_SME2p3.

Type: `uint8_t`

Default value: 0

**sve.sme\_highest\_implemented\_priority**

When SME Priority Control and SME2p2 are implemented, controls the highest implemented priority.

Type: `uint8_t`

Default value: 0

**sve.sme\_only**

If SME is implemented, whether SVE functionality is available only when SM=1.

Type: `bool`

Default value: `false`

**sve.sme\_ssve\_fp8\_support\_level**

If FEAT\_SME2 and FEAT\_FP8 are implemented, whether FP8 operations are supported in Streaming Mode where not implemented outside Streaming Mode. Possible values are: 0 - No support above FEAT\_FP8, 1 - FEAT\_SSVE\_FP8FMA, 2 - FEAT\_SSVE\_FP8DOT4, 3 - FEAT\_SSVE\_FP8DOT2.

Type: `uint8_t`

Default value: 0

**sve.sme\_veclebs\_implemented**

Which SME vector lengths are implemented. Represented as a bitfield where `bit[n]==1` implies SME vector length of  $128 \cdot 2^n$  bits is implemented.

Type: `uint8_t`

Default value: 7

**sve.smidr\_el1\_implementer\_val**

The value of SMIDR\_EL1.Implementer.

Type: `uint8_t`

Default value: 65

**sve.smidr\_el1\_nsmc\_val**

The value of SMIDR\_EL1.NSMC.

Type: `uint8_t`

Default value: 0

**sve.smidr\_el1\_revision\_val**

The value of SMIDR\_EL1.Revision.

Type: `uint8_t`

Default value: 0

**sve.smidr\_el1\_sh\_val**

The value of SMIDR\_EL1.SH.

Type: `uint8_t`

Default value: 0

**sve.sve2\_version**

The version of SVE2 if implemented. Possible values are: 0 - FEAT\_SVE2, 1 - FEAT\_SVE2p1, 2 - FEAT\_SVE2p2, 3 - FEAT\_SVE2p3.

Type: `uint8_t`

Default value: 0

**sve.sve\_dabt\_far\_behaviour**

Whether the FAR reported on a Data Abort is imprecise. Possible values are: 0 - FAR Precise, 1 - FAR not Precise on abort due to an SVE contiguous vector load/store or an SME load/store, 2 - As per 1, but only for predicated SVE/SME instructions, 3 - As per 1, but only for predicated SME/SVE load/store instructions that are executed in Streaming Mode.

Type: `uint8_t`

Default value: 0

**sve.sve\_wp\_far\_behaviour**

FAR reporting behavior on a Watchpoint debug exception. Possible values are: 0 - FAR Precise, 1 - FAR not Precise on abort due to an SVE contiguous vector load/store in Streaming SVE mode, or an SME load/store, 2 - FAR not valid on abort due to an SVE contiguous vector load/store in Streaming SVE mode, or an SME load/store, 3 - As per 1, but only for predicated SVE/SME instructions, 4 - As per 1, but only for predicated SME/SVE load/store instructions that are executed in Streaming Mode.

Type: `uint8_t`

Default value: 0

**sve.trace\_za\_tilewise**

Whether tile-wise accesses to ZA are traced tile-wise rather than array-wise. Note: if false, column-wise accesses cause an event for every vector in the tile.

Type: `bool`

Default value: `true`

**sve.undef\_invalid\_combined\_movprfx**

If a combined MOVPRFX is invalid, raise an UNDEF exception. Otherwise, **NOP** the second half. This parameter is deprecated.

Type: `bool`

Default value: `true`

**sve.unknown\_value**

Simulated value for a state that has an **UNKNOWN** value after reset.

Type: `uint64_t`

Default value: `0xdeaddeaddeaddead`

**sve.vecLEN**

SVE vector length in units of 64 bits.

Type: `uint8_t`

Default value: `8`

**sve.z\_reg\_on\_load\_fault\_behaviour**

Governs the behavior of destination Z-registers in case of a load fault. Possible values are: 0 - Register becomes **UNKNOWN**, 1 - Register is preserved.

Type: `uint8_t`

Default value: `0`

**sve.za\_on\_svl\_increase\_behaviour**

Controls the state of the previously inaccessible portion of the ZA registers on SVL increase. Possible values are: 0 - Retain values, 1 - Zero ZA.

Type: `uint8_t`

Default value: `0`

**sve.za\_tag\_checked**

Whether memory accesses due to SME LDR and STR instructions that access the SME ZA array perform a Tag Check.

Type: `bool`

Default value: `true`

**swp\_with\_xzr\_is\_st\_atomic**

If true, swp with dest as xzr is treated as store atomic.

Type: `bool`

Default value: `true`

**sync\_ext\_abort\_is\_sync\_serror**

Treat synchronous external aborts as synchronous SErrors (RASv8.9). 0, synchronous external abort. 1, synchronous serror.

Type: `bool`

Default value: `false`

**system\_pmu\_id**

When FEAT\_SPMU is implemented, indicates the largest value `s` to select a System PMU `<s>`.

Type: `uint8_t`

Default value: 0

**take\_ccfail\_tsc\_trap**

When `take_ccfail_undef=1` this parameter controls whether or not an SMC instruction that is trapped by HCR\_EL2.TSC but fails its condition code check generates a trap to EL2.

Type: `bool`

Default value: `false`

**take\_ccfail\_undef**

UNDEF exception is taken even if condition code check fails.

Type: `bool`

Default value: `true`

**tcr\_ps\_reserved\_value\_size**

Physical size treated when TCR.(I)PS is programmed with a reserved value. 0, 48 bits. 1, 52 bits. The parameter value is treated 0 if LPA is not supported.

Type: `uint8_t`

Default value: 0

**tcr\_tgx\_bit1\_stateful**

TCR.TGx[1] is stateful even without 16k granule support.

Type: `bool`

Default value: `false`

### **tcr\_txsz\_undersize\_should\_fault**

If large VA is not supported, Whether undersized TxSZ value should generate translation fault.

Type: `bool`

Default value: `true`

### **tdosa\_traps\_osdlr\_if\_no\_os\_double\_lock**

MDCR\_EL\*.TDOSA enables trap on OSDLR\_EL1 and DBGOSDLR when OS double-lock is not implemented.

Type: `bool`

Default value: `true`

### **tidcp\_traps\_el0\_undef\_imp\_def**

TIDCP has priority over UNDEF for accesses to **IMPLEMENTATION DEFINED** functionality from ELO.

Type: `bool`

Default value: `true`

### **tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

### **tlbi\_or\_ic\_invalid\_xt**

Behavior of TLBI and IC instructions that don't take Xt as an argument when Xt != 0b11111. 0: TLB and IC not UNDEF, 1: TLBI UNDEF, IC not UNDEF, 2: TLBI not UNDEF, IC UNDEF, 3: TLBI and IC UNDEF.

Type: `uint8_t`

Default value: `0`

### **tlbi\_stall\_enabled**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: `bool`

Default value: `false`

**tlbid\_nis**

Number of bits supported for IS TLBI Domains (FEAT\_TLBID). 0 - 16 : log(number of supported ISH tlbid domains).

Type: `uint8_t`

Default value: 0

**tlbid\_nos**

Number of bits supported for OS TLBI Domains (FEAT\_TLBID). 0 - 16 : log(number of supported OSH tlbid domains).

Type: `uint8_t`

Default value: 0

**tlbid\_nvis**

Number of bits supported for Virtual IS TLBI Domains (FEAT\_TLBID). 0 - 5 : log(number of supported Virtual ISH tlbid domains).

Type: `uint8_t`

Default value: 0

**tlbid\_nvos**

Number of bits supported for Virtual OS TLBI Domains (FEAT\_TLBID). 0 - 5 : log(number of supported Virtual OSH tlbid domains).

Type: `uint8_t`

Default value: 0

**tps\_support\_level**

Support for thread private state extension: 0 - Not implemented. 1 - Implemented at ELO only (FEAT\_TPS). 2 - Implemented at ELO, EL1 and EL2. (FEAT\_TPS, FEAT\_TPSP).

Type: `uint8_t`

Default value: 0

**trace\_full\_simd\_reg\_with\_nep**

Whether full simd register is traced even if partial update is done when FPCR.NEP=1.

Type: `bool`

Default value: `false`



**trace\_has\_sysreg\_access**

ETM trace registers support access via system registers.

Type: `bool`

Default value: `true`

**trace\_icc\_registers\_as\_icv\_when\_redirected**

If true, update trace with ICV, instead of ICC when ICV registers are accessed depending on the core state.

Type: `bool`

Default value: `false`

**trace\_physical\_registers\_when\_host\_virtualisation\_enabled**

Trace sysreg accesses to physical register accessed (0=disabled, 1=Trace only ELR/SPSR\_EL1 as ELR/SPSR\_EL2, 2=Trace all redirected registers as physical registers affecting all features inc. the host virtualised registers.

Type: `uint8_t`

Default value: `0`

**trace\_xzr\_in\_core\_regs64\_trace**

Whether CORE\_REGS64\_READ traces XZR and WZR input registers.

Type: `bool`

Default value: `true`

**trap\_dc\_cmo\_to\_pou\_if\_nop**

Whether traps to DC CMO operations to PoU are ignored if the same is treated as **NOP**.

Type: `bool`

Default value: `true`

**trap\_ic\_cmo\_to\_pou\_if\_nop**

Whether traps to IC CMO operations to PoU are ignored if the same is treated as **NOP**.

Type: `bool`

Default value: `true`

**trap\_reserved\_group3\_id\_regs**

Whether setting HCR\_EL2.TID3 traps reserved group3 id registers.

Type: `bool`

Default value: `false`

### **trbe\_cmod**

TRBE Customer Modified.

Type: `uint8_t`

Default value: `0`

### **trbe\_des**

Designer, JEP106 identification code.

Type: `uint16_t`

Default value: `0x0`

### **trbe\_external\_abort\_handling**

Describes how the PE manages External aborts on writes made by the Trace Buffer Unit to the trace buffer. (0->External abort is reported to TRBE. From Armv9.3, the value 0 is not permitted and will be 1 if Armv9.3 is implemented. 1-> External abort is ignored. 2->The External abort generates an SError and the error is not reported to TRBE.).

Type: `uint8_t`

Default value: `0`

### **trbe\_has\_hardware\_translation\_table\_update**

If true, address translation performed by the Trace Buffer Extension manages the Access Flag and dirty state.

Type: `bool`

Default value: `true`

### **trbe\_implemented\_version**

Trace Buffer Extension implemented version, 1: FEAT\_TRBE implemented (Armv9.0), 2: FEAT\_TRBEv1p1 and FEAT\_TRBE\_EXC are implemented.

Type: `uint8_t`

Default value: `1`

### **trbe\_mpam**

TRBE MPAM support.

Type: `uint8_t`

Default value: 0

**trbe\_part**

Part number.

Type: uint16\_t

Default value: 0x0

**trbe\_partid\_max**

Largest permitted TRBDEVID1.PARTID value.

Type: uint16\_t

Default value: 0x0

**trbe\_pmg\_max**

Largest permitted TRBDEVID1.PMG value if FEAT\_MPAMv2 is implemented, otherwise 0xff.

Type: uint16\_t

Default value: 0x0

**trbe\_revand**

TRBE component minor revision.

Type: uint8\_t

Default value: 0

**trbe\_revision**

TRBE architecture revision.

Type: uint8\_t

Default value: 0

**trbe\_stop\_on\_misaligned\_pointers**

If true, the Trace Buffer Extension will stop tracing if a buffer pointer is not aligned to TRBIDR\_EL1.Align.

Type: bool

Default value: false

**trbe\_trbptr\_el1\_has\_res0**

If true, when TRBIDR\_EL1.Align is not zero, bits [M-1:0] in TRBPTR\_EL1 are not stateful.

Type: `bool`

Default value: `false`

#### **`treat-dcache-cmos-to-occ-as-nop`**

Implement CMOs to Outer cache level as **NOP**.

Type: `bool`

Default value: `false`

#### **`treat-dcache-cmos-to-poc-as-nop`**

Whether dcache maintenance operations to the point of coherency are required for instruction to data coherence. 0 - Clean/Invalidate ops required, 1 - Clean/Invalidate ops not required and cannot generate faults, 2 - Clean/Invalidate ops not required but can generate faults.

Type: `uint8_t`

Default value: 0

#### **`treat-dcache-cmos-to-pou-as-nop`**

Whether dcache invalidation to the point of unification is required for instruction to data coherence. 0 - Invalidate ops required, 1 - Invalidate ops not required and cannot generate faults, 2 - Invalidate ops not required but can generate faults.

Type: `uint8_t`

Default value: 0

#### **`treat-dcache-invalidate-as-clean-invalidate`**

Treat data cache invalidate operations as clean and invalidate.

Type: `bool`

Default value: `false`

#### **`treat-icache-cmos-to-pou-as-nop`**

If `has_coherent_icache` is true, whether instruction cache invalidation operations to PoU which are treated as **NOP** can generate fault. 0 - cannot generate faults, 1 - can generate faults.

Type: `uint8_t`

Default value: 0

#### **`treat_PAC_as_NOP`**

Non-architecture parameter. Treat Pointer Authentication as **NOP**. When the parameter is true behaviour of FEAT\_PAAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions

are **NOP**. Branch, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAAuth traps.

Type: `bool`

Default value: `false`

#### **`treat_forced_normal_as_device_for_excl_atomics`**

Whether exclusive/atomic access is supported in same manner as access to device if stage1 is Device memory and final memory attribute forced to normal by FWB.

Type: `bool`

Default value: `false`

#### **`treat_pld_as_nop`**

If true, treat PLD as **NOP**.

Type: `bool`

Default value: `false`

#### **`treat_pli_as_nop`**

If true, treat PLI as **NOP**.

Type: `bool`

Default value: `false`

#### **`treat_wfi_wfe_as_nop`**

If true, never go into wait state for WFI or WFE instructions.

Type: `bool`

Default value: `false`

#### **`truncate_pc_on_illegal_exception_return_to_aarch32`**

On Illegal ERET to AArch32, truncate PC to 32-bits.

Type: `bool`

Default value: `true`

#### **`ttt_fetch_fault_report_type`**

Behaviour for TTT fetch faults: 0, report as data abort. 1, report as instruction abort.

Type: `bool`

Default value: `false`

### **tune\_spe\_cache\_events**

"Set the percentage of cache event bits set on a load instruction. Supplying any valid, non-empty JSON makes PMSNEVFR\_EL1[23:19] stateful. Each field if present, must be between 5-95. Omitted fields default to 0. Example: {'l2\_dcache\_access':20, 'l2\_dcache\_miss':20, 'dcache\_modified':20, 'recently\_fetched':20, 'data\_snooped':20}"

Type: `string`

Default value: `N/A`

### **undef\_ccsidr2\_access\_for\_unimplemented\_aarch32**

Whether access to CCSIDR2 is undef if AArch32 is implemented or not at EL1.

Type: `bool`

Default value: `false`

### **unification-level**

Level of Unification Inner Shareable for the cache hierarchy.

Type: `uint8_t`

Default value: `1`

### **unification-uniprocessor-level**

Level of Unification Uniprocessor for the cache hierarchy.

Type: `uint8_t`

Default value: `1`

### **unpred\_LSE128\_overlap**

Constrained unpredictable behaviours for 128-bit LSE overlap. 1 Constraint\_UNDEF, 2 Constraint\_NOP.

Type: `uint8_t`

Default value: `1`

### **unpred\_brb\_iall\_or\_inj\_invalid\_xt\_behave\_as\_undef**

If true, BRB IALL/INJ instruction will behave as **UNDEFINED** if Xt != 0b11111.

Type: `bool`

Default value: `false`

**unpred\_brbe\_next\_branch\_cycle\_count\_unknown**

If true, cycle count value for the next BRBE branch record after BRB INJ execution outside prohibited region is unknown.

Type: `bool`

Default value: `false`

**unpred\_clear\_ISV\_for\_exception\_before\_software\_step**

Whether ESR\_ELx.ISV bit is cleared/set, when it is constrained unpredictable due to a different exception before a software step exception.

Type: `bool`

Default value: `false`

**unpred\_edscr\_ns\_set\_unknown\_bit**

Unknown(x) bit in NS field in EDSCR can be configure to 0 or 1.

Type: `bool`

Default value: `false`

**unpred\_edscr\_rw\_unknown\_bits\_read\_as\_1**

Unknown(x) bits in RW field in EDSCR are read as 1 instead of 0.

Type: `bool`

Default value: `false`

**unpred\_edscr\_status\_read\_as\_no\_syndrome**

Controls the choice of EDSCR.STATUS bit-values, when it is constrained unpredictable behaviour due to a different exception before a halting step debug event.

Type: `bool`

Default value: `false`

**unpred\_extdbg\_unknown\_bits**

Data used to fill only in **UNKNOWN** bit-fields of external debug registers e.g., EDPFR and EDDFR.

Type: `uint64_t`

Default value: `0x0`

**unpred\_load\_single\_reg\_overlap\_with\_wb**

Constrained unpredictable behaviours for single load with writeback(might impact certain load pair instructions) 0 Constraint\_WBSUPPRESS, 1 Constraint\_UNDEF, 2 Constraint\_NOP.

Type: `uint8_t`

Default value: 0

**unpred\_mrsmsr\_currentlymapped\_undef**

**UNPREDICTABLE** register access (accessible from current mode using different instruction) modeled as **NOP** when false and **UNDEF** when true.

Type: `bool`

Default value: `false`

**unpred\_mrsmsr\_protfailed\_undef**

**UNPREDICTABLE** register access (not accessible from current PL and security state) modeled as **NOP** when false and **UNDEF** when true.

Type: `bool`

Default value: `false`

**unpred\_mte\_stzgm\_tag\_operation\_before\_data**

Whether Tag operations are performed before data operations for an STGZM instruction.

Type: `bool`

Default value: `true`

**unpred\_mte\_tag\_read\_when\_ata\_controls\_are\_zero\_or\_untagged\_attr**

Constrained unpredictable for MTE tag read when ATA controls are 0 or untagged attribute. false, Read as zero. true, Permitted to generate an external abort if a read of data from the same address would generate an external abort.

Type: `bool`

Default value: `false`

**unpred\_mte\_tag\_store\_data\_cache\_instr\_to\_device\_mem\_as\_alignment\_fault**

Constrained unpredictable choice for MTE instructions which store tags (on DC instructions) to memory locations marked as Device. false, Storing the data, if any, to the locations. true, Generating an Alignment Fault.

Type: `bool`

Default value: `false`



**unpred\_mte\_tag\_store\_to\_device\_mem\_as\_alignment\_fault**

Constrained unpredictable choice for STZGM instruction which store tags to memory locations marked as Device. false, Storing the data, if any, to the locations. true, Generating an Alignment Fault.

Type: `bool`

Default value: `false`

**unpred\_nested\_virtualization\_nv\_behaviour**

Constrained unpredictable choices for HCR\_EL2.NV=0 and HCR\_EL2.NV1=1 with respect to nested virtualization- 0, Behave as defined in the specification as per bit values- 1, Behave as if HCR\_EL2.NV=1 and HCR\_EL2.NV1=1 for all purpose other than reading back HCR\_EL2.NV- 2, Behave as if HCR\_EL2.NV=0 and HCR\_EL2.NV1=0 for all purpose other than reading back HCR\_EL2.NV1.

Type: `uint8_t`

Default value: 0

**unpred\_par\_attr\_returns\_mair**

If true, PAR\_EL1.ATTR represents the memory attributes as per the MAIR value instead of the ones in the descriptor.

Type: `bool`

Default value: `false`

**unpred\_poe2\_va\_ress\_mismatch**

Constrained unpredictable choices when bits marked as RESS do not all have the same value for POE2 registers containing a VA - 0, Generating a translation abort on use of the register- 1, Reserved sign extended bits of the register are same as bit[52] or bit[48] based on if large VA is supported or not, for all purposes other than reading back the register- 2, Reserved sign extended bits of the register are same as bit[52] or bit[48] based on if large VA is supported or not, for all purposes .

Type: `uint8_t`

Default value: 0

**unpred\_s2\_hw\_dirty\_update\_on\_atomic\_wo\_read\_perm\_fault**

Constrained unpredictable behavior for atomic instructions that generate a stage 2 permission fault only due to lack of read permission, on a stage 2 writable-clean descriptor. If true, hardware is allowed to update the stage 2 dirty state; else, the dirty update is suppressed.

Type: `bool`

Default value: `false`

**unpred\_sctlr\_c\_0\_taggable\_behaviour**

Controls unpredictable effects when SCTLTR\_ELx.C=0 for a stage 1 translation regime on whether memory is treated as taggable. Values: 0=Tagged, 1=Untagged but forced to Tagged when FWB=1 and stage 2 restores WB, 2 = Untagged.

Type: uint8\_t

Default value: 2

**unpred\_store\_exclusive\_base\_overlap**

Constrained unpredictable behaviours for store exclusive when s==n. 0 Constraint\_NONE, 1 Constraint\_UNDEF, 2 Constraint\_NOP.

Type: uint8\_t

Default value: 0

**unpred\_store\_pair\_and\_single\_reg\_overlap\_with\_wb**

Constrained unpredictable behaviours for pair and single store with writeback(doesn't cover store exclusive) 0 Constraint\_NONE, 1 Constraint\_UNDEF, 2 Constraint\_NOP.

Type: uint8\_t

Default value: 0

**unpred\_tchange\_tenter\_and\_texit\_behaviour**

TCHANGE, TENTER and TEXTIT unpredictable behaviour in debug state. 0, **NOP**. 1, Undefined. 2, Execute as in non-debug state.

Type: uint8\_t

Default value: 0

**unpred\_tlbi\_not\_in\_monitor\_mode**

Constrained unpredictable behaviors for AArch32 TLBI instructions executed in secure privileged mode other than Monitor mode. 0: Preferred behavior (default), 1: UNDEF, 2: **NOP**, 3: execute as if had been executed in Monitor mode.

Type: uint8\_t

Default value: 0

**unpred\_tps\_range**

When TPMINn\_ELx > TMAXn\_ELx: 0 - Prevent all accesses. 1 - Allow all accesses. 2 - Wrap around.

Type: uint8\_t

Default value: 0

#### **unpred\_tps\_va\_ress\_mismatch**

Constrained unpredictable choices when bits marked as RESS do not all have the same value for TPS registers containing a VA - 0, Generating a translation abort on use of the register- 1, Reserved sign extended bits of the register are same as bit[52] or bit[48] based on if large VA is supported or not, for all purposes other than reading back the register- 2, Reserved sign extended bits of the register are same as bit[52] or bit[48] based on if large VA is supported or not, for all purposes .

Type: `uint8_t`

Default value: 0

#### **unpred\_tsize\_aborts**

Behaviour when TSize is out of range. 0, force into range. 1, translation fault, forces `unpred_tsize_pamax_aborts` to 1.

Type: `bool`

Default value: `false`

#### **unpred\_tsize\_pamax\_aborts**

Behaviour when stage 2 TSize exceeds the physical address size (or 40bits, from AArch32). 0, force into range. 1, translation fault. Ignored if `unpred_tsize_aborts` is 1.

Type: `bool`

Default value: `false`

#### **unpred\_vnocr\_el2\_ress\_mismatch**

Constrained unpredictable choices when bits marked as RESS do not all have the same value for VNCR\_EL2 - 0, Generating an EL2 translation regime translation abort on use of the VNCR\_EL2 register- 1, Reserved sign extended bits of VNCR\_EL2 are same as bit[52] or bit[48] based on if large VA is supported or not, for all purposes other than reading back the register- 2, Reserved sign extended bits of VNCR\_EL2 are same as bit[52] or bit[48] based on if large VA is supported or not, for all purposes .

Type: `uint8_t`

Default value: 0

#### **unpred\_zero\_spsr\_btype**

Constrained unpredictable control to make SPSR\_ELx.BTYPE 0 instead of PSTATE.BTYPE on synchronous exceptions other than Software Step, PC alignment fault, Instruction Abort, Breakpoint or Address Matching Vector Catch, Illegal Execution State, BRK instruction, Branch Target.

Type: `bool`

Default value: `true`

### **`unpredictable_exclusive_abort_memtype`**

Cause MMU abort if exclusive access is not supported in certain memory type (0=exclusives allowed in all memory types, 1=exclusives abort in Device memory types, 2=exclusives abort in any type other than WB inner cacheable).

Type: `uint8_t`

Default value: 0

### **`unpredictable_hvc_behaviour`**

HVC unpredictable behaviour. 0, UNDEF. 1, **NOP**.

Type: `uint8_t`

Default value: 0

### **`unpredictable_smc_behaviour`**

SMC unpredictable behaviour. 0, UNDEF. 1, **NOP**.

Type: `uint8_t`

Default value: 0

### **`unpredictable_wfet_and_wfit_behaviour`**

WFET and WFIT unpredictable behaviour in debug state. 0, **UNDEFINED**. 1, **NOP**.

Type: `uint8_t`

Default value: 1

### **`unsupported_atomic_fault_type`**

Type of fault reported on unsupported atomic access. 0 = external abort if any reported by interconnect, 1 = precise unsupported atomic fault, 2 = precise external abort, 3 = imprecise external abort.

Type: `uint8_t`

Default value: 0

### **`unsupported_fp8_format_behaviour`**

Behaviour when FPMR.{F8S1,F8S2,F8D} are programmed to a reserved value 0->FP8 Inputs are treated as a signalling NaN, FP8 outputs are 0xFF 1->Format is treated as FPMR.{F8S1,F8S2,F8D} & 0x1.

Type: `uint8_t`

Default value: 0

### **unsupported\_hw\_update\_fault\_type**

Type of abort reported when hw update to descriptor is done using unsupported memtype (0=No abort, 1=IMPDEF abort caused by memtype, 2=Sync external abort).

Type: `uint8_t`

Default value: 0

### **use\_architectural\_names**

Use names SP/LR/PC instead of R13/R14/R15.

Type: `bool`

Default value: `false`

### **use\_sif\_to\_compute\_pan**

Where FEAT\_PAN3 is implemented, whether SCR\_EL3.SIF bit is used to determine instruction access permission for the purpose of PAN.

Type: `bool`

Default value: `false`

### **use\_stage1\_sh\_as\_input\_to\_stage2**

IMPDEF case of whether to use stage1 shareability or OuterShareable as input to stage2 if stage1 is Device memtype.

Type: `bool`

Default value: `false`

### **use\_tlb\_contig\_hint**

Translation table entries with the contiguous hint bit set generate large TLB entries.

Type: `bool`

Default value: `false`

### **user\_defined\_rom\_table\_debug\_power\_config**

User defined ROM Table debug power domains for ED,CTI,PMU and TRACE, and DBGPCR configuration. The "version" field and "cores" array are mandatory. The "dbgpcr" array, if provided, must contain unique integers in the range [0, 31] describing which debug power domains have power control implemented. The "rom" and "dbgpcr" fields in objects in the "cores" array are only allowed when 'debug\_rom\_is\_flat' is false. All power domain ID fields ("rom", "ed/pmu", "cti", "etm") must be in the range [0, 31]. The "ed/pmu" field is mandatory. Example JSON for a hierarchical

debug ROM layout: '{"version": 0, "dbgpcr": [0, 1], "cores":[{"dbgpcr": [1, 31], "rom": 0, "ed/pmu": 0, "cti": 31, "etm": 1}, {"ed/pmu": 0}]}'.

Type: `string`

Default value: `N/A`

### **vmte\_support\_level**

Specify the Virtual Tagging Base Feature support level:- 0 Not Implemented- 1 Implement Virtual Tagging and Checking controls and instructions, but not enablement (FEAT\_VMTE+FEAT\_VMTETC)- 2 As per 1 but controls to enable Virtual tagging and tag checking are not ignored, and access to tags in memory is enabled (FEAT\_VMTEE+FEAT\_VMTETCE).

Type: `uint8_t`

Default value: `0`

### **vpu\_datapath\_width**

VPU data path width.

Type: `uint8_t`

Default value: `128`

### **walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

### **warn\_for\_dbgwcr\_reserved\_values\_with\_razwi\_bits**

Display a warning when DBGWCR is programmed with a reserved value even if some bits(e.g. HMC) are **RAZ/WI**.

Type: `bool`

Default value: `true`

### **warn\_unpredictable\_in\_v7**

If true, behaviour which is unpredictable in V7 yet is predictable in V8 will produce a warning.

Type: `bool`

Default value: `true`

**watchpoint-log2secondary\_restriction**

log2 size of secondary restriction of FAR/EDWAR possible values on watchpoint hit for load/store operations.

Type: `uint8_t`

Default value: 0

**wfe\_wakeup\_delay**

Configure WFE wakeup delay in CPU cycles.

Type: `uint32_t`

Default value: 0x0

**wfi\_wakeup\_delay**

Configure WFI wakeup delay in CPU cycles.

Type: `uint32_t`

Default value: 0x0

**wnr\_is\_read\_for\_s2f\_on\_s1\_atomic\_instr\_fault**

Whether WnR is 0 for stage2 fault on stage1 for atomic instructions.

Type: `bool`

Default value: `false`

**wnr\_is\_read\_for\_s2f\_on\_s1\_dbm\_update**

Whether WnR is 0 for stage2 fault on stage1 descriptor dbm update.

Type: `bool`

Default value: `false`

**wp\_ignores\_dbm\_update**

If true, dbm update is ignored on watchpoint hit.

Type: `bool`

Default value: `false`

**wp\_num\_reporting**

When reporting of the watchpoint number on Watchpoint Exceptions and Debug Events is performed - When FEAT\_Debugv8p9 is implemented or otherwise required - When FEAT\_Debugv8p9 or FEAT\_SME is implemented.

Type: `uint8_t`

Default value: 0

## 3.5 AHCI\_SATA

Defined in `LISA/AHCI_SATA.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### About AHCI\_SATA

This component implements an AHCI controller including attached SATA disks. It connects as a PCIe end-point device to a PCIe framework.

### Iris and MTI instances for AHCI\_SATA

This model has the following Iris instances:

Name	Instance type
AHCI_SATA	<a href="#">AHCI_SATA</a>
AHCI_SATA.ahci_master	<a href="#">PVBUSMaster</a>
AHCI_SATA.register_slave	<a href="#">PVBUSSlave</a>

This model has the following MTI trace components:

Name	Component type
AHCI_SATA	<a href="#">AHCI_SATA</a>
AHCI_SATA.ahci_master	<a href="#">PVBUSMaster</a>
AHCI_SATA.register_slave	<a href="#">PVBUSSlave</a>

### Ports for AHCI\_SATA

Port	Direction	Protocol	Description
ahci_dma_m	master	<a href="#">PVBUS</a>	AHCI device performs DMA accesses via master
client_s	slave	<a href="#">PCIDevice2ClientProtocol</a>	PCIDevice client slave port, used for MSI-X
pvbuss	slave	<a href="#">PVBUS</a>	AHCI pci/control/config/status registers



## Parameters for AHCI\_SATA

### **force\_mode**

Force disk to report support for at most PIO/DMA/NCQ mode (only for testing/bring-up purposes). PIO mode is always supported. Use NCQ for maximum performance (default).

Type: `string`

Default value: `"NCQ"`

### **image\_path**

Comma separated list of zero or more disk images (up to 32). Each image represents one SATA disk which is connected to one port of the AHCI controller. Empty list elements are allowed and result in a SATA port which has no disk attached. Empty string (default) means: One SATA port with no disk attached. Use `'truncate -s 4T disk.img'` to create a 4 TByte sparse image. Use `'dd if=/dev/zero of=disk.img bs=1M count=42'` to create a 42 MByte non-sparse image.

Type: `string`

Default value: `N/A`

### **run\_async**

Do host I/O in a background thread asynchronously. Enabling this makes the simulation non-deterministic and may or may not improve performance. Default is `'false'` (do all disk accesses synchronously).

Type: `bool`

Default value: `false`

## 3.6 AMBAPV2PVBus

Defined in `examples/SystemCEExport/Bridges/AMBAPV2PVBus.lisa`.

### Changes in 11.31.15

The following parameters were added:

- `mpam-mec-attribute-transform`

### About AMBAPV2PVBus

- PVBus does not support transactions with `byte_enable` set (strobing transactions, in AXI terms). This bridge component rejects them. The only exception is when `byte_enable` is used to signify that the transaction is a tag-only write request (no data).
- Variants of this component also exist with multiple input and output ports.

The bridge enables the ACP port to treat a transaction as coherent. It provides an additional parameter to specify the default shared bit value for incoming AMBA-PV transactions.

It also enables the shared bit to be specified by the “shareable” attribute of an AMBA-PV transaction using the `amba_pv_attributes` class. (Requires you to define the `AMBA_PV_INCLUDE_ATTRIBUTES` macro at compile time.)

### Limitations

- The debug channel does not support all MTE operations, it only supports tag-only requests (for reading or writing tags only).

### Iris and MTI instances for AMBAPV2PVBUS

This model has the following Iris instances:

Name	Instance type
<code>AMBAPV2PVBUS</code>	<code>AMBAPV2PVBUS</code>
<code>AMBAPV2PVBUS.bus_master</code>	<code>PVBUSMaster</code>

This model has the following MTI trace components:

Name	Component type
<code>AMBAPV2PVBUS</code>	<code>AMBAPV2PVBUS</code>
<code>AMBAPV2PVBUS.bus_master</code>	<code>PVBUSMaster</code>

### Ports for AMBAPV2PVBUS

Port	Direction	Protocol	Description
<code>amba_pv_s</code>	slave	<code>AMBAPV</code>	-
<code>pvbuss_m</code>	master	<code>PVBUS</code>	-

### Parameters for AMBAPV2PVBUS

#### **base\_addr**

Base address.

Type: `uint64_t`

Default value: `0x0`

#### **mpam-mec-attribute-transform**

User-defined transform describing how MPAM Attributes and/or MEC ID are encoded into PVBUS attributes such as ManagerID, ExtendedID or UserFlags. For example, 'ExtendedID[62:55]=MPAM\_PMG, ExtendedID[54:39]=MPAM\_PARTID, ExtendedID[38]=MPAM\_SP[0], ExtendedID[37]=MPAM\_SP[1], UserFlags[31:16]=MECID'. An empty string disables MPAM/MEC support.

Type: `string`

Default value: ""

### **report\_errors**

Report transactions which do not comply with PVBUS protocol requirements.

Type: `bool`

Default value: `false`

### **shareable**

Shareable default.

Type: `bool`

Default value: `true`

## 3.7 AMBAPV2PVBUSx4

Defined in `examples/SystemCEExport/Bridges/AMBAPV2PVBUSx4.lisa`.

### About AMBAPV2PVBUSx4

AMBA-PV to PVBUS protocol converter with array size 4.

### Iris and MTI instances for AMBAPV2PVBUSx4

This model has the following Iris instances:

Name	Instance type
AMBAPV2PVBUSx4	AMBAPV2PVBUSx4
AMBAPV2PVBUSx4.ambapv2pvbus_U (where U = 0-3)	AMBAPV2PVBUS
AMBAPV2PVBUSx4.ambapv2pvbus_U.bus_master (where U = 0-3)	PVBUSMaster

This model has the following MTI trace components:

Name	Component type
AMBAPV2PVBUSx4.ambapv2pvbus_U (where U = 0-3)	AMBAPV2PVBUS
AMBAPV2PVBUSx4.ambapv2pvbus_U.bus_master (where U = 0-3)	PVBUSMaster

### Ports for AMBAPV2PVBUSx4

Port	Direction	Protocol	Description
amba_pv_s	slave	AMBAPV	From SystemC, bridge to array port pvbus_m[x].
pvbus_m	master	PVBUS	To SystemC.

### Parameters for AMBAPV2PVBUSx4

This component does not have any parameters.

## 3.8 AMBAPV2PVBusx8

Defined in `examples/SystemCEExport/Bridges/AMBAPV2PVBusx8.lisa`.

### About AMBAPV2PVBusx8

AMBA-PV to PVBus protocol converter with array size 8.

### Iris and MTI instances for AMBAPV2PVBusx8

This model has the following Iris instances:

Name	Instance type
AMBAPV2PVBusx8	AMBAPV2PVBusx8
AMBAPV2PVBusx8.ambapv2pvbus_U (where $U = 0-7$ )	AMBAPV2PVBus
AMBAPV2PVBusx8.ambapv2pvbus_U.bus_master (where $U = 0-7$ )	PVBusMaster

This model has the following MTI trace components:

Name	Component type
AMBAPV2PVBusx8.ambapv2pvbus_U (where $U = 0-7$ )	AMBAPV2PVBus
AMBAPV2PVBusx8.ambapv2pvbus_U.bus_master (where $U = 0-7$ )	PVBusMaster

### Ports for AMBAPV2PVBusx8

Port	Direction	Protocol	Description
amba_pv_s	slave	AMBAPV	From SystemC, bridged to pvbus_m[x].
pvbus_m	master	PVBus	To SystemC.

### Parameters for AMBAPV2PVBusx8

This component does not have any parameters.

## 3.9 AMBAPVACE2PVBus

Defined in `examples/SystemCEExport/Bridges/AMBAPVACE2PVBus.lisa`.

### Changes in 11.31.15

The following parameters were added:

- `mpam-mec-attribute-transform`
- `separate-rw-dmi-invalidate`

### About AMBAPVACE2PVBus

- AMBAPVACE2PVBus depends on the AMBA-PV API, which must be at least version 1.4.

- The translation of bus transactions by the bridge has some impact on performance. Bus masters that cache memory transactions avoid much of this impact. The bridge does not support DMI.
- PVBUS does not support transactions with `byte_enable` set (strobing transactions, in AXI terms). This bridge component rejects them. The only exception is used to signify that the transaction is a tag-only write request (no data).

## Limitations

- The debug channel does not support all MTE operations, it only supports tag-only requests (for reading or writing tags only).

## Iris and MTI instances for AMBAPVACE2PVBUS

This model has the following Iris instances:

Name	Instance type
AMBAPVACE2PVBUS	AMBAPVACE2PVBUS
AMBAPVACE2PVBUS.bus_master	PVBUSMaster

This model has the following MTI trace components:

Name	Component type
AMBAPVACE2PVBUS	AMBAPVACE2PVBUS
AMBAPVACE2PVBUS.bus_master	PVBUSMaster

## Ports for AMBAPVACE2PVBUS

Port	Direction	Protocol	Description
amba_pv_ace_s	slave	AMBAPVACE	-
pvbuss_m	master	PVBUS	-

## Parameters for AMBAPVACE2PVBUS

### **mpam-mec-attribute-transform**

User-defined transform describing how MPAM Attributes and/or MEC ID are encoded into PVBUS attributes such as ManagerID, ExtendedID or UserFlags. For example, 'ExtendedID[62:55]=MPAM\_PMG, ExtendedID[54:39]=MPAM\_PARTID, ExtendedID[38]=MPAM\_SP[0], ExtendedID[37]=MPAM\_SP[1], UserFlags[31:16]=MECID'. An empty string disables MPAM/MEC support.

Type: `string`

Default value: ""

### **report\_errors**

Report transactions which do not comply with PVBUS protocol requirements.

Type: `bool`

Default value: `false`

**separate-rw-dmi-invalidate**

Transport separate read or write DMI invalidation upstream.

Type: `bool`

Default value: `true`

### 3.10 AMBAPVSignal2SGSignal

Defined in `examples/SystemCEexport/Bridges/AMBAPVSignal2SGSignal.lisa`.



Variants of this component also exist with multiple input and output ports.

#### Iris and MTI instances for AMBAPVSignal2SGSignal

This model has the following Iris instances:

Name	Instance type
AMBAPVSignal2SGSignal	AMBAPVSignal2SGSignal

#### Ports for AMBAPVSignal2SGSignal

Port	Direction	Protocol	Description
<code>amba_pv_signal_s</code>	slave	AMBAPVSignal	Input slave port for connection from top-level AMBAPVSignal slave port.
<code>sg_signal_m</code>	master	Signal	Handles outgoing signal state changes. Converted signal state changes are sent out through this port.

#### Parameters for AMBAPVSignal2SGSignal

This component does not have any parameters.

### 3.11 AMBAPVSignal2SGSignalx1024

Defined in `examples/SystemCEexport/Bridges/AMBAPVSignal2SGSignalx1024.lisa`.

#### About AMBAPVSignal2SGSignalx1024

AMBA-PV Signal to SystemGenerator Signal array protocol converter.

## Iris and MTI instances for AMBAPVSignal2SGSignalx1024

This model has the following Iris instances:

Name	Instance type
AMBAPVSignal2SGSignalx1024	AMBAPVSignal2SGSignalx1024

### Ports for AMBAPVSignal2SGSignalx1024

Port	Direction	Protocol	Description
amba_pv_signal_s	slave	AMBAPVSignal	-
sg_signal_m	master	Signal	-

### Parameters for AMBAPVSignal2SGSignalx1024

This component does not have any parameters.

## 3.12 AMBAPVSignal2SGSignalx16

Defined in `examples/SystemCEExport/Bridges/AMBAPVSignal2SGSignalx16.lisa`.

### About AMBAPVSignal2SGSignalx16

AMBA-PV Signal to SystemGenerator Signal array protocol converter.

## Iris and MTI instances for AMBAPVSignal2SGSignalx16

This model has the following Iris instances:

Name	Instance type
AMBAPVSignal2SGSignalx16	AMBAPVSignal2SGSignalx16

### Ports for AMBAPVSignal2SGSignalx16

Port	Direction	Protocol	Description
amba_pv_signal_s	slave	AMBAPVSignal	Input slave port for connection from top-level AMBAPVSignal slave port.
sg_signal_m	master	Signal	Handles outgoing signal state changes. Converted signal state changes are sent out through this port.

### Parameters for AMBAPVSignal2SGSignalx16

This component does not have any parameters.

### 3.13 AMBAPVSignal2SGSignalx224

Defined in `examples/SystemCEExport/Bridges/AMBAPVSignal2SGSignalx224.lisa`.

#### About AMBAPVSignal2SGSignalx224

AMBA-PV Signal to SystemGenerator Signal array protocol converter.

#### Iris and MTI instances for AMBAPVSignal2SGSignalx224

This model has the following Iris instances:

Name	Instance type
AMBAPVSignal2SGSignalx224	AMBAPVSignal2SGSignalx224

#### Ports for AMBAPVSignal2SGSignalx224

Port	Direction	Protocol	Description
amba_pv_signal_s	slave	AMBAPVSignal	-
sg_signal_m	master	Signal	-

#### Parameters for AMBAPVSignal2SGSignalx224

This component does not have any parameters.

### 3.14 AMBAPVSignal2SGSignalx4

Defined in `examples/SystemCEExport/Bridges/AMBAPVSignal2SGSignalx4.lisa`.

#### About AMBAPVSignal2SGSignalx4

AMBA-PV Signal to SystemGenerator Signal array protocol converter.

#### Iris and MTI instances for AMBAPVSignal2SGSignalx4

This model has the following Iris instances:

Name	Instance type
AMBAPVSignal2SGSignalx4	AMBAPVSignal2SGSignalx4

#### Ports for AMBAPVSignal2SGSignalx4

Port	Direction	Protocol	Description
amba_pv_signal_s	slave	AMBAPVSignal	-
sg_signal_m	master	Signal	-

#### Parameters for AMBAPVSignal2SGSignalx4

This component does not have any parameters.



## 3.15 AMBAPVSignal2SGSignalx48

Defined in `examples/SystemCEExport/Bridges/AMBAPVSignal2SGSignalx48.lisa`.

### About AMBAPVSignal2SGSignalx48

AMBA-PV Signal to SystemGenerator Signal array protocol converter.

### Iris and MTI instances for AMBAPVSignal2SGSignalx48

This model has the following Iris instances:

Name	Instance type
AMBAPVSignal2SGSignalx48	<a href="#">AMBAPVSignal2SGSignalx48</a>

### Ports for AMBAPVSignal2SGSignalx48

Port	Direction	Protocol	Description
amba_pv_signal_s	slave	<a href="#">AMBAPVSignal</a>	-
sg_signal_m	master	<a href="#">Signal</a>	-

### Parameters for AMBAPVSignal2SGSignalx48

This component does not have any parameters.

## 3.16 AMBAPVSignal2SGSignalx8

Defined in `examples/SystemCEExport/Bridges/AMBAPVSignal2SGSignalx8.lisa`.

### About AMBAPVSignal2SGSignalx8

AMBA-PV Signal to SystemGenerator Signal array protocol converter.

### Iris and MTI instances for AMBAPVSignal2SGSignalx8

This model has the following Iris instances:

Name	Instance type
AMBAPVSignal2SGSignalx8	<a href="#">AMBAPVSignal2SGSignalx8</a>

This model has the following MTI trace components:

Name	Component type
AMBAPVSignal2SGSignalx8	<a href="#">AMBAPVSignal2SGSignalx8</a>

### Ports for AMBAPVSignal2SGSignalx8

Port	Direction	Protocol	Description
amba_pv_signal_s	slave	AMBAPVSignal	-
sg_signal_m	master	Signal	-

#### Parameters for AMBAPVSignal2SGSignalx8

This component does not have any parameters.

## 3.17 AMBAPVSignal2SGSignalx960

Defined in `examples/SystemCEExport/Bridges/AMBAPVSignal2SGSignalx960.lisa`.

#### About AMBAPVSignal2SGSignalx960

AMBA-PV Signal to SystemGenerator Signal array protocol converter.

#### Iris and MTI instances for AMBAPVSignal2SGSignalx960

This model has the following Iris instances:

Name	Instance type
AMBAPVSignal2SGSignalx960	AMBAPVSignal2SGSignalx960

### Ports for AMBAPVSignal2SGSignalx960

Port	Direction	Protocol	Description
amba_pv_signal_s	slave	AMBAPVSignal	-
sg_signal_m	master	Signal	-

#### Parameters for AMBAPVSignal2SGSignalx960

This component does not have any parameters.

## 3.18 AMBAPVSignal2SGSignalx988

Defined in `examples/SystemCEExport/Bridges/AMBAPVSignal2SGSignalx988.lisa`.

#### About AMBAPVSignal2SGSignalx988

AMBA-PV Signal to SystemGenerator Signal array protocol converter.

#### Iris and MTI instances for AMBAPVSignal2SGSignalx988

This model has the following Iris instances:

Name	Instance type
AMBAPVSignal2SGSignalx988	AMBAPVSignal2SGSignalx988

### Ports for AMBAPVSignal2SGSignalx988

Port	Direction	Protocol	Description
amba_pv_signal_s	slave	AMBAPVSignal	-
sg_signal_m	master	Signal	-

#### Parameters for AMBAPVSignal2SGSignalx988

This component does not have any parameters.

## 3.19 AMBAPVSignalState2SGStateSignal

Defined in `examples/SystemCEExport/Bridges/AMBAPVSignalState2SGStateSignal.lisa`.



Variants of this component also exist with multiple input and output ports.

#### Iris and MTI instances for AMBAPVSignalState2SGStateSignal

This model has the following Iris instances:

Name	Instance type
AMBAPVSignalState2SGStateSignal	AMBAPVSignalState2SGStateSignal

### Ports for AMBAPVSignalState2SGStateSignal

Port	Direction	Protocol	Description
amba_pv_signal_s	slave	AMBAPVSignalState	-
sg_signal_m	master	StateSignal	-

#### Parameters for AMBAPVSignalState2SGStateSignal

This component does not have any parameters.

## 3.20 AMBAPVSignalState2SGStateSignalx4

Defined in `examples/SystemCEExport/Bridges/AMBAPVSignalState2SGStateSignalx4.lisa`.

#### About AMBAPVSignalState2SGStateSignalx4

AMBA-PV SignalState to SystemGenerator StateSignal protocol converter.

#### Iris and MTI instances for AMBAPVSignalState2SGStateSignalx4

This model has the following Iris instances:

Name	Instance type
AMBAPVSignalState2SGStateSignalx4	AMBAPVSignalState2SGStateSignalx4

### Ports for AMBAPVSignalState2SGStateSignalx4

Port	Direction	Protocol	Description
amba_pv_signal_s	slave	AMBAPVSignalState	-
sg_signal_m	master	StateSignal	-

### Parameters for AMBAPVSignalState2SGStateSignalx4

This component does not have any parameters.

## 3.21 AMBAPVValue2SGValue

Defined in `examples/SystemCEExport/Bridges/AMBAPVValue2SGValue.lisa`.



Note

Variants of this component also exist with multiple input and output ports.

### Iris and MTI instances for AMBAPVValue2SGValue

This model has the following Iris instances:

Name	Instance type
AMBAPVValue2SGValue	AMBAPVValue2SGValue

### Ports for AMBAPVValue2SGValue

Port	Direction	Protocol	Description
amba_pv_value_s	slave	AMBAPVValue	-
sg_value_m	master	Value	-

### Parameters for AMBAPVValue2SGValue

This component does not have any parameters.

## 3.22 AMBAPVValue2SGValue64

Defined in `examples/SystemCEExport/Bridges/AMBAPVValue2SGValue64.lisa`.



Variants of this component also exist with multiple input and output ports.

### Iris and MTI instances for AMBAPVValue2SGValue64

This model has the following Iris instances:

Name	Instance type
AMBAPVValue2SGValue64	AMBAPVValue2SGValue64

### Ports for AMBAPVValue2SGValue64

Port	Direction	Protocol	Description
amba_pv_value_s	slave	AMBAPVValue64	-
sg_value_m	master	Value_64	-

### Parameters for AMBAPVValue2SGValue64

This component does not have any parameters.

## 3.23 AMBAPVValue2SGValue64x4

Defined in `examples/SystemCEExport/Bridges/AMBAPVValue2SGValue64x4.lisa`.

### About AMBAPVValue2SGValue64x4

AMBA-PV Value64 to SystemGenerator Value\_64 array protocol converter.

### Iris and MTI instances for AMBAPVValue2SGValue64x4

This model has the following Iris instances:

Name	Instance type
AMBAPVValue2SGValue64x4	AMBAPVValue2SGValue64x4

### Ports for AMBAPVValue2SGValue64x4

Port	Direction	Protocol	Description
amba_pv_value_s	slave	AMBAPVValue64	-
sg_value_m	master	Value_64	-

### Parameters for AMBAPVValue2SGValue64x4

This component does not have any parameters.

## 3.24 AMBAPVValue2SGValuex4

Defined in `examples/SystemCEExport/Bridges/AMBAPVValue2SGValuex4.lisa`.

### About AMBAPVValue2SGValuex4

AMBA-PV Value to SystemGenerator Value array protocol converter.

### Iris and MTI instances for AMBAPVValue2SGValuex4

This model has the following Iris instances:

Name	Instance type
AMBAPVValue2SGValuex4	AMBAPVValue2SGValuex4

### Ports for AMBAPVValue2SGValuex4

Port	Direction	Protocol	Description
amba_pv_value_s	slave	AMBAPVValue	-
sg_value_m	master	Value	-

### Parameters for AMBAPVValue2SGValuex4

This component does not have any parameters.

## 3.25 AMBAPVValue642SMMUv3AEMIdentify

Defined in `examples/SystemCEExport/Bridges/AMBAPVValue642SMMUv3AEMIdentify.lisa`.

### About AMBAPVValue642SMMUv3AEMIdentify

AMBA-PV Value64 to SMMUv3AEMIdentify protocol converter.

### Iris and MTI instances for AMBAPVValue642SMMUv3AEMIdentify

This model has the following Iris instances:

Name	Instance type
AMBAPVValue642SMMUv3AEMIdentify	AMBAPVValue642SMMUv3AEMIdentify

### Ports for AMBAPVValue642SMMUv3AEMIdentify

Port	Direction	Protocol	Description
identify_reply	master	AMBAPVValue64	-
identify_request	slave	AMBAPVValue64	-

Port	Direction	Protocol	Description
identify	master	SMMUv3AEMIdentifyProtocol	-

### Parameters for AMBAPVValue642SMMUv3AEMIdentify

This component does not have any parameters.

## 3.26 AMBAPVValue642VECB

Defined in `examples/SystemCEExport/Bridges/AMBAPVValue642VECB.lisa`.

### About AMBAPVValue642VECB

AMBA-PV to VECB protocol converter.

### Iris and MTI instances for AMBAPVValue642VECB

This model has the following Iris instances:

Name	Instance type
AMBAPVValue642VECB	AMBAPVValue642VECB

### Ports for AMBAPVValue642VECB

Port	Direction	Protocol	Description
amba_pv_ctrl_s	slave	AMBAPVValue	-
amba_pv_data_s	slave	AMBAPVValue64	-
vecb_m	master	VECBProtocol	-

### Parameters for AMBAPVValue642VECB

This component does not have any parameters.

## 3.27 AMBAPVValueState2SGValueState

Defined in `examples/SystemCEExport/Bridges/AMBAPVValueState2SGValueState.lisa`.



Note

Variants of this component also exist with multiple input and output ports.

### Iris and MTI instances for AMBAPVValueState2SGValueState

This model has the following Iris instances:

Name	Instance type
AMBAPVValueState2SGValueState	AMBAPVValueState2SGValueState

### Ports for AMBAPVValueState2SGValueState

Port	Direction	Protocol	Description
amba_pv_value_s	slave	AMBAPVValueState	-
sg_value_m	master	ValueState	-

### Parameters for AMBAPVValueState2SGValueState

This component does not have any parameters.

## 3.28 AMBAPVValueState2SGValueState64

Defined in `examples/SystemCEExport/Bridges/AMBAPVValueState2SGValueState64.lisa`.



Note

Variants of this component also exist with multiple input and output ports.

### Iris and MTI instances for AMBAPVValueState2SGValueState64

This model has the following Iris instances:

Name	Instance type
AMBAPVValueState2SGValueState64	AMBAPVValueState2SGValueState64

### Ports for AMBAPVValueState2SGValueState64

Port	Direction	Protocol	Description
amba_pv_value_s	slave	AMBAPVValueState64	-
sg_value_m	master	ValueState_64	-

### Parameters for AMBAPVValueState2SGValueState64

This component does not have any parameters.

## 3.29 AMBAPVValueState2SGValueState64x4

Defined in `examples/SystemCEExport/Bridges/AMBAPVValueState2SGValueState64x4.lisa`.

### About AMBAPVValueState2SGValueState64x4

AMBA-PV ValueState64 to SystemGenerator ValueState\_64 array protocol converter.



## Iris and MTI instances for AMBAPVValueState2SGValueState64x4

This model has the following Iris instances:

Name	Instance type
AMBAPVValueState2SGValueState64x4	AMBAPVValueState2SGValueState64x4

## Ports for AMBAPVValueState2SGValueState64x4

Port	Direction	Protocol	Description
amba_pv_value_s	slave	AMBAPVValueState64	-
sg_value_m	master	ValueState_64	-

## Parameters for AMBAPVValueState2SGValueState64x4

This component does not have any parameters.

## 3.30 AMBAPVValueState2SGValueStatex4

Defined in `examples/SystemCEExport/Bridges/AMBAPVValueState2SGValueStatex4.lisa`.

### About AMBAPVValueState2SGValueStatex4

AMBA-PV ValueState to SystemGenerator ValueState array protocol converter.

## Iris and MTI instances for AMBAPVValueState2SGValueStatex4

This model has the following Iris instances:

Name	Instance type
AMBAPVValueState2SGValueStatex4	AMBAPVValueState2SGValueStatex4

## Ports for AMBAPVValueState2SGValueStatex4

Port	Direction	Protocol	Description
amba_pv_value_s	slave	AMBAPVValueState	-
sg_value_m	master	ValueState	-

## Parameters for AMBAPVValueState2SGValueStatex4

This component does not have any parameters.

## 3.31 ARMAEMv8MCT

Defined in `LISA/ARMAEMv8MCT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
v8.0M	Full support
v8.1M	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

## About ARMAEMv8MCT

ARMAEMv8MCT CPU component.

## Iris and MTI instances for ARMAEMv8MCT

This model has the following Iris instances:

Name	Instance type
ARMAEMv8MCT	ARM_AEMv8M
ARMAEMv8MCT.acp_mapper	PVBusMapper
ARMAEMv8MCT.ext_bus	PVBusLogger
ARMAEMv8MCT.ext_bus.mapper	PVBusMapper
ARMAEMv8MCT.l1_incoherent_interconnect	PVCache
ARMAEMv8MCT.l1_incoherent_interconnect.downstream[0].pvbusmaster	PVBusMaster
ARMAEMv8MCT.l1_incoherent_interconnect.upstream[Z] (where Z = 0-17)	PVBusSlave
ARMAEMv8MCT.l1dcache	PVCache
ARMAEMv8MCT.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMAEMv8MCT.l1dcache.upstream[0]	PVBusSlave
ARMAEMv8MCT.l1licache	PVCache
ARMAEMv8MCT.l1licache.downstream[0].pvbusmaster	PVBusMaster
ARMAEMv8MCT.l1licache.upstream[0]	PVBusSlave
ARMAEMv8MCT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

Name	Component type
ARMAEMv8MCT	ARM_AEMv8M
ARMAEMv8MCT.acp_mapper	PVBusMapper
ARMAEMv8MCT.ext_bus	PVBusLogger
ARMAEMv8MCT.ext_bus.mapper	PVBusMapper
ARMAEMv8MCT.l1_incoherent_interconnect	PVCache
ARMAEMv8MCT.l1_incoherent_interconnect.downstream[0].pvbusmaster	PVBusMaster
ARMAEMv8MCT.l1_incoherent_interconnect.upstream[Z] (where Z = 0-17)	PVBusSlave
ARMAEMv8MCT.l1dcache	PVCache
ARMAEMv8MCT.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMAEMv8MCT.l1dcache.upstream[0]	PVBusSlave
ARMAEMv8MCT.l1licache	PVCache
ARMAEMv8MCT.l1licache.downstream[0].pvbusmaster	PVBusMaster

Name	Component type
ARMAEMv8MCT.l1licache.upstream[0]	PVBusSlave
ARMAEMv8MCT.l2_flusher	AsyncCacheFlushUnit

## Ports for ARMAEMv8MCT

Port	Direction	Protocol	Description
ahbd	slave	PVBus	Debug AHB - core bus slave driven by the DAP.
ahbp_m	master	PVBus	The core will generate Vendor System data accesses on this port.
ahbs	slave	PVBus	External master (e.g. DMA) can write TCMs (whether or not enabled in xTCMCR).
auxfault	slave	Value	This is wired to the Auxiliary Fault Status Register.
bigend	slave	Signal	Configure big endian data format.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
coproc_bus	slave	CoprocBusProtocol	Co-Processor Interface
cpuwait	slave	Signal	When this signal is HIGH out of reset, it forces the processor into a quiescent state that delays its boot-up sequence and instruction execution until this signal is driven LOW.
currpri	master	Value	Current execution priority.
dap_s	slave	PVBus	Debug Access Port (DAP).
dbgen	slave	Signal	Invasive debug enable.
dbgrestart	slave	Signal	External debug request.
dbgrestarted	master	Signal	External debug request.
edbgrq	slave	Signal	External debug request.
etm_reset	slave	Signal	Separate reset for ETM, if param "has_etm_reset" is true.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE and corresponds to the RTL TXEV and RXEV signals.
fpxxc	master	Value	Port that sends the value of the FPxxC exception flags (FPIXC, FPIDC, FPOFC, FPUFC, FPDZC, FPIOC). Each flag is present in the same position as in the FPSCR register (e.g. fpxxc[0] = FPIOC).
halted	master	Signal	External debug request.
idau_invalidate_region	slave	Value_64	64 bit number to invalid IDAU memory range (start_address<<32 end_address)
idau	master	PVBus	The core will generate IDAU Bus request.
initpahben	slave	Signal	Enable P-AHB on the next reset
initvtor_ns	slave	Value	Reset configuration port - Non-Secure Vector table offset (VTOR.TBLOFF[31:7]) out of reset This port remains functional no matter ARMv8-M Security Extensions are included or not When ARMv8-M Security Extensions are not included, all exceptions will use NS vector base address given by this port.
initvtor_s	slave	Value	Reset configuration port - Secure Vector table offset (VTOR.TBLOFF[31:7]) out of reset It becomes functional when ARMv8-M Security Extensions are included When ARMv8-M Security Extensions are not included, this port will be ignored.
intisr	slave	Signal	This signal array delivers signals to the NVIC.
intnmi	slave	Signal	Configure non maskable interrupt.

Port	Direction	Protocol	Description
intnum	master	Value	Exception number of the current execution context (from IPSR[8:0]) When the processor is in Thread mode, INTNUM is 0 When the processor is in Handler mode, INTNUM is the exception number of the currently-executing exception.
io_port_in	slave	PVBus	I/O port pair. See the documentation for the io_port_out port.
io_port_out	master	PVBus	I/O port pair. Used if IOP is true. Transactions from io_port_out which do not "match" should be returned via io_port_in. For performance reasons, the I/O port interface is not modelled directly. Instead, a simple PVBus gasket is inserted at the point in the memory system where the I/O port would be. In hardware, a device would be attached to the port which would tell the CPU whether it would like to intercept each transaction, given its address. This can be modelled in a performant manner by connecting a PVBusMapper-based device to io_port_out which intercepts transactions of interest and passes other transactions back to the CPU via io_port_in. Your I/O port device model is also responsible for aborting transactions which would be aborted on hardware (e.g. exclusives) if necessary.
lockdcaic	slave	Signal	Disable access to the Direct Cache Access Instruction Cache (DCAIC) registers.
locknsmpu	slave	Signal	Disable writes to the Non-Secure MPU_*_NS registers
locknsvtor	slave	Signal	Disable writes to VTOR_NS
lockpahb	slave	Signal	P-AHB related ports Disable writes to PAHBCR
locksau	slave	Signal	Disable writes to the SAU_* registers
locksmpu	slave	Signal	Disable writes to the Secure MPU_* registers
locksvtaircr	slave	Signal	Disable writes to VTOR_S, AIRCR.PRIS, AIRCR.BFHFNMINS
lockup	master	Signal	Asserted when the processor is in lockup state.
niden	slave	Signal	Non-invasive debug enable.
poreset	slave	Signal	Raising this signal will do a power-on reset of the core.
pv_ppbus_m	master	PVBus	The core will generate External Private Peripheral Bus requests on this port.
pdbus_m	master	PVBus	The core will generate bus requests on this port.
sleepdeep	master	Signal	Asserted when the processor is in deep sleep.
sleeping	master	Signal	Asserted when the processor is in sleep.
spiden	slave	Signal	Secure invasive debug enable.
spniden	slave	Signal	Secure non-invasive debug enable.
stcalib	slave	Value	These are the secure and non-secure calibration values for the SysTick timer. stcalib[0] is the secure timer calibration, stcalib[1] is the non-secure timer calibration. In the case where the Security Extension is not present, stcalib[0] should be used. The following bit flag is modelled for each stcalib[*][25] - NOREF - Indicates no reference clock integrated. If stclk is not in use, set to 1
stclk	slave	ClockSignal	This is the reference clock for the SysTick timer.
sysreset	slave	Signal	Raising this signal will put the core into reset mode (but does not reset the debug logic).
sysresetreq	master	Signal	Asserted to indicate that a reset is required.
ticks	master	InstructionCount	Port allowing the number of instructions since startup to be read from the CPU.

## Parameters for ARMAEMv8MCT

### SAU\_REGIONX.BADDR

Base address of SAU region0 at reset.

Type: `uint32_t`

Default value: `0x0`

#### **SAU\_REGIONX.ENABLE**

Enable SAU region0 at reset.

Type: `bool`

Default value: `false`

#### **SAU\_REGIONX.LADDR**

Limit address of SAU region0 at reset.

Type: `uint32_t`

Default value: `0x0`

#### **SAU\_REGIONX.NSC**

Set NSC for SAU region0 at reset.

Type: `bool`

Default value: `false`

#### **AFSR\_type**

0:No AFSR, 1:state-only 2:Many bits set from (im)precise aborts on AXI, TCM, etc.

Type: `uint8_t`

Default value: `1`

#### **AIRCR.BFHFNMINReset**

If true, set the bit after reset (as if by IMP\_DEF mechanism). Ignored if SECEXT=false.

Type: `bool`

Default value: `false`

#### **AIRCR.BFHFNMINWritable**

Is AIRCR.BFHFNMIN bit[13] writeable.

Type: `bool`

Default value: `true`

#### **AIRCR.ENDIANNESS**

Initialize processor to big endian mode.

Type: `bool`

Default value: `false`

#### **AIRCR.PRIS\_writable**

Is AIRCR.PRIS bit[14] writeable.

Type: `bool`

Default value: `true`

#### **AIRCR.VECTCLRACTIVE\_changes\_mode**

Asserting AIRCR.VECTCLRACTIVE clears IPSR and any active exceptions. The mode is also changed to thread if this flag is true. Ignored for v8-M.

Type: `bool`

Default value: `true`

#### **AIRCR\_NS.DIT\_reset**

If true and AIRCR\_NS.DIT\_writable==0, set the bit after reset (as if by IMP\_DEF mechanism).

Type: `bool`

Default value: `true`

#### **AIRCR\_NS.DIT\_writable**

Is AIRCR\_NS.DIT bit[4] writeable.

Type: `bool`

Default value: `true`

#### **AIRCR\_S.DIT\_reset**

If true and AIRCR\_S.DIT\_writable==0, set the bit after reset (as if by IMP\_DEF mechanism).

Type: `bool`

Default value: `true`

#### **AIRCR\_S.DIT\_writable**

Is AIRCR\_S.DIT bit[4] writeable.

Type: `bool`

Default value: `true`

**BB\_PRESENT**

Enable bitbanding.

Type: `bool`

Default value: `false`

**BEATS\_PER\_TICK**

Number of beats from each in-flight vector instruction executed in 1 tick (1,2 or 4).

Type: `uint8_t`

Default value: 2

**BF\_is\_nop**

BF instruction executes as **NOP**, even if we have `LO_BRANCH_INFO`.

Type: `bool`

Default value: `true`

**CCR.BP**

Reset value of the Configuration and Control Register's branch prediction enable bit.

Type: `bool`

Default value: `true`

**CCR.BP\_writable**

Whether it is possible to modify the Configuration and Control Register's branch prediction enable bit.

Type: `bool`

Default value: `false`

**CDEMAPPEDONCP**

Bit N specifies whether the instruction for coprocessor N (CP7:CP0) is redirected to the CDE module.

Type: `uint8_t`

Default value: 255

**CDERTLID**

Value of `ID_AFR0.CDERTLID`.

Type: `uint8_t`

Default value: 32

### **CFGMEMALIAS**

Memory address alias bit for the ITCM, DTCM and P-AHB regions. 0=No alias, 1=Alias bit 24, 2=Alias bit 25, 4=Alias bit 26, 8=Alias bit 27, 16=Alias bit 28.

Type: `uint8_t`

Default value: 0

### **CFGNOCDECP**

Bit N means external coprocessor N (CP7:CP0) disable for CDE coprocessor.

Type: `uint8_t`

Default value: 0

### **CFGPAHBSZ**

Size of the P-AHB peripheral port memory region. 0=P-AHB disabled, 1=64MB, 2=128MB, 3=256MB, 4=512MB.

Type: `uint8_t`

Default value: 0

### **CPNSPRESENT**

Bit N means external coprocessor N (CP7:CP0) is accessible in Non-Secure state.

Type: `uint8_t`

Default value: 255

### **CPSPRESENT**

Bit N means external coprocessor N (CP7:CP0) is accessible in Secure state.

Type: `uint8_t`

Default value: 255

### **CPUID**

Set SCS CPUID Base Register. If set to zero, a default CPUID is used.

Type: `uint32_t`

Default value: 0x0



**CTI**

CTI (Cross Trigger Interface) included.

Type: `bool`

Default value: `false`

**CTI\_irq0\_pin**

CTI interrupt request 0 pin.

Type: `uint16_t`

Default value: 4

**CTI\_irq1\_pin**

CTI interrupt request 1 pin.

Type: `uint16_t`

Default value: 5

**DTGU**

DTCM Security Gate Unit included.

Type: `bool`

Default value: `false`

**DTGUBLKSZ**

DTCM gate unit block size. Size= $\text{pow}(2, \text{DTGUBLKSZ} + 5)$  bytes.

Type: `uint8_t`

Default value: 3

**DTGUMAXBLKS**

Maximum number of DTCM gate unit blocks. Number of blocks= $\text{pow}(2, \text{DTGUMAXBLKS})$ .

Type: `uint8_t`

Default value: 0

**DWT\_CTRL.NOCYCCNT**

DWT cycle-counter not present (v8M-bl/v6M never have one).

Type: `bool`

Default value: `false`

**DWT\_CTRL.NOPRFCNT**

DWT performance-counters not present (v8M-bl/v6M never have them).

Type: `bool`

Default value: `false`

**DWT\_CTRL.NUMCOMP**

Number of watchpoint unit comparators implemented.

Type: `uint8_t`

Default value: 4

**DWT\_DEVARCH.REVISION**

0: V2, 1: V2.1.

Type: `uint8_t`

Default value: 1

**DWT\_FUNCTION0.ID**

Sets the capabilities of the comparator that is accessible via the register, DWT\_FUNCTION0. If 'baseline' is set, invalid ID bits are cleared.

Type: `uint8_t`

Default value: 11

**DWT\_FUNCTION1.ID**

Sets the capabilities of the comparator that is accessible via the register, DWT\_FUNCTION1. If 'baseline' is set, invalid ID bits are cleared.

Type: `uint8_t`

Default value: 30

**DWT\_FUNCTION10.ID**

Sets the capabilities of the comparator that is accessible via the register, DWT\_FUNCTION10. If 'baseline' is set, invalid ID bits are cleared.

Type: `uint8_t`

Default value: 11

**DWT\_FUNCTION11.ID**

Sets the capabilities of the comparator that is accessible via the register, DWT\_FUNCTION11. If 'baseline' is set, invalid ID bits are cleared.

Type: uint8\_t

Default value: 30

**DWT\_FUNCTION12.ID**

Sets the capabilities of the comparator that is accessible via the register, DWT\_FUNCTION12. If 'baseline' is set, invalid ID bits are cleared.

Type: uint8\_t

Default value: 11

**DWT\_FUNCTION13.ID**

Sets the capabilities of the comparator that is accessible via the register, DWT\_FUNCTION13. If 'baseline' is set, invalid ID bits are cleared.

Type: uint8\_t

Default value: 30

**DWT\_FUNCTION14.ID**

Sets the capabilities of the comparator that is accessible via the register, DWT\_FUNCTION14. If 'baseline' is set, invalid ID bits are cleared.

Type: uint8\_t

Default value: 11

**DWT\_FUNCTION15.ID**

Sets the capabilities of the comparator that is accessible via the register, DWT\_FUNCTION15. If 'baseline' is set, invalid ID bits are cleared.

Type: uint8\_t

Default value: 30

**DWT\_FUNCTION2.ID**

Sets the capabilities of the comparator that is accessible via the register, DWT\_FUNCTION2. If 'baseline' is set, invalid ID bits are cleared.

Type: uint8\_t

Default value: 11

**DWT\_FUNCTION3.ID**

Sets the capabilities of the comparator that is accessible via the register, DWT\_FUNCTION3. If 'baseline' is set, invalid ID bits are cleared.

Type: uint8\_t

Default value: 30

**DWT\_FUNCTION4.ID**

Sets the capabilities of the comparator that is accessible via the register, DWT\_FUNCTION4. If 'baseline' is set, invalid ID bits are cleared.

Type: uint8\_t

Default value: 11

**DWT\_FUNCTION5.ID**

Sets the capabilities of the comparator that is accessible via the register, DWT\_FUNCTION5. If 'baseline' is set, invalid ID bits are cleared.

Type: uint8\_t

Default value: 30

**DWT\_FUNCTION6.ID**

Sets the capabilities of the comparator that is accessible via the register, DWT\_FUNCTION6. If 'baseline' is set, invalid ID bits are cleared.

Type: uint8\_t

Default value: 11

**DWT\_FUNCTION7.ID**

Sets the capabilities of the comparator that is accessible via the register, DWT\_FUNCTION7. If 'baseline' is set, invalid ID bits are cleared.

Type: uint8\_t

Default value: 30

**DWT\_FUNCTION8.ID**

Sets the capabilities of the comparator that is accessible via the register, DWT\_FUNCTION8. If 'baseline' is set, invalid ID bits are cleared.

Type: uint8\_t

Default value: 11

**DWT\_FUNCTION9.ID**

Sets the capabilities of the comparator that is accessible via the register, DWT\_FUNCTION9. If 'baseline' is set, invalid ID bits are cleared.

Type: `uint8_t`

Default value: 30

**DWT\_TRACE**

Support for DWT trace, controls the DWT\_CTRL.NOTRCPKT bit. false : No DWT trace included, true: DWT trace included.

Type: `bool`

Default value: `true`

**DWT\_VMASK\_reset\_data**

DWT\_VMASK register reset value.

Type: `uint32_t`

Default value: 0x0

**ERRDEVID.NUM**

RAS: Number of implemented error record indexes, 0 to 56.

Type: `uint8_t`

Default value: 56

**FPB\_HAS\_LSR**

FPB has LAR and LSR for software lock if mainline.

Type: `bool`

Default value: `true`

**FP\_CTRL.NUM\_CODE**

Number of breakpoint unit comparators implemented (limited to 15 in V6M or baseline).

Type: `uint8_t`

Default value: 8

**FP\_CTRL.NUM\_LIT**

How many Literals FPB supports remapping (ignored if baseline or TZM).

Type: `uint8_t`

Default value: 0

#### **FP\_REMAP.RMPSPT**

FPB supports remapping (ignored if baseline or SECEXT).

Type: `bool`

Default value: `true`

#### **ID\_DFR0.Debug\_Model\_M\_profile**

Set whether debug extensions are implemented.

Type: `bool`

Default value: `true`

#### **ID\_ISAR0.CmpBranch**

Support for Compare and Branch instructions. 1 = Supports CBNZ and CBZ instructions; 3 = Supports non-predicated low overhead looping (WLS, DLS, LE, and LC) and branch future (BF, BFX, BFL, BFLX, and BFCSEL) instructions.

Type: `uint8_t`

Default value: 3

#### **ID\_ISAR0.coproc\_instrs**

Supported Coprocessor instructions 0: None 1: CDP, LDC, MCR, MRC, and STC instructions 2: As for 1, and CDP2, LDC2, MCR2, MRC2, and STC2 instructions 3: As for 2, and MCRR and MRRC instructions 4: As for 2, and MCRR and MRRC instructions.

Type: `uint8_t`

Default value: 4

#### **ID\_ISAR1.extend\_instrs**

level of support for extend instructions.

Type: `uint8_t`

Default value: 2

#### **ID\_ISAR1.interwork\_instrs**

level of support for Interworking instructions.

Type: `uint8_t`

Default value: 2

**ID\_ISAR2.MultiAccessInt**

level of support for interruptible multi-access instructions.

Type: uint8\_t

Default value: 2

**ID\_ISAR2.multS\_instrs**

level of support for advanced signed Multiply instructions.

Type: uint8\_t

Default value: 3

**ID\_ISAR2.multU\_instrs**

level of support for advanced unsigned Multiply instructions.

Type: uint8\_t

Default value: 2

**ID\_ISAR3.SIMD\_instrs**

level of support for SIMD instructions.

Type: uint8\_t

Default value: 3

**ID\_ISAR3.saturate\_instrs**

level of support for saturate instructions.

Type: uint8\_t

Default value: 1

**ID\_ISAR3.synchprim\_instrs**

level of support for synchronization primitives ID\_ISAR3.

Type: uint8\_t

Default value: 1

**ID\_ISAR4.synchPrim\_instrs\_frac**

level of support for synchronization primitives ID\_ISAR4.

Type: uint8\_t

Default value: 3

**ID\_ISAR4.unpriv\_instrs**

supported unprivileged instructions 0: None 1: LDRBT, LDRT, STRBT, and STRT instructions 2: As for 1, and LDRHT, LDRSBT, LDRSHT, and STRHT instructions.

Type: `uint8_t`

Default value: 2

**ID\_ISAR4.withshifts\_instrs**

level of support for instructions with shifts.

Type: `uint8_t`

Default value: 3

**ID\_ISAR5.PACBTI**

0: PAC/BTI not implemented, 1: PAC implemented using the QARMA5 algorithm with BTI, 2: PAC implemented using an IMP DEF algorithm with BTI, 4: PAC implemented using the QARMA3 algorithm with BTI.

Type: `uint8_t`

Default value: 0

**ID\_MMFR0.Auxiliary\_registers**

Auxiliary registers bits in ID\_MMFR0, indicate the support for Auxiliary registers.

Type: `bool`

Default value: `true`

**ID\_MMFR0.Outermost\_shareability**

Outermost shareability bits in ID\_MMFR0, indicate the outermost shareability domain implemented.

Type: `uint8_t`

Default value: 0

**ID\_MMFR0.ShareLvl**

Shareability levels bits in ID\_MMFR0, indicate the number of Shareability levels implemented.

Type: `uint8_t`

Default value: 1



**INITPAHBEN**

The P-AHB enable state at reset.

Type: `bool`

Default value: `false`

**INITVTOR\_NS**

Non-Secure vector-table offset at reset.

Type: `uint32_t`

Default value: `0x0`

**INITVTOR\_S**

Secure vector-table offset at reset.

Type: `uint32_t`

Default value: `0x0`

**IOP**

Send all d-side transactions to the port, `io_port_out`. Transactions which do not match should be returned to the port, `io_port_in`.

Type: `bool`

Default value: `false`

**IRQDIS0**

IRQ line disable mask. Bit `n` of this 32-bit parameter disables `IRQ[n+0]`.

Type: `uint32_t`

Default value: `0x0`

**IRQDIS1**

IRQ line disable mask. Bit `n` of this 32-bit parameter disables `IRQ[n+32]`.

Type: `uint32_t`

Default value: `0x0`

**IRQDIS10**

IRQ line disable mask. Bit `n` of this 32-bit parameter disables `IRQ[n+320]`.

Type: `uint32_t`

Default value: 0x0

**IRQDIS11**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+352].

Type: uint32\_t

Default value: 0x0

**IRQDIS12**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+384].

Type: uint32\_t

Default value: 0x0

**IRQDIS13**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+416].

Type: uint32\_t

Default value: 0x0

**IRQDIS14**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+448].

Type: uint32\_t

Default value: 0x0

**IRQDIS15**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+480].

Type: uint32\_t

Default value: 0x0

**IRQDIS2**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+64].

Type: uint32\_t

Default value: 0x0

**IRQDIS3**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+96].

Type: uint32\_t

Default value: 0x0

**IRQDIS4**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+128].

Type: uint32\_t

Default value: 0x0

**IRQDIS5**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+160].

Type: uint32\_t

Default value: 0x0

**IRQDIS6**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+192].

Type: uint32\_t

Default value: 0x0

**IRQDIS7**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+224].

Type: uint32\_t

Default value: 0x0

**IRQDIS8**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+256].

Type: uint32\_t

Default value: 0x0

**IRQDIS9**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+288].

Type: uint32\_t

Default value: 0x0

**ITGU**

ITCM Security Gate Unit included.

Type: bool

Default value: `false`

**ITGUBLKSZ**

ITCM gate unit block size.  $\text{Size} = \text{pow}(2, \text{ITGUBLKSZ} + 5)$  bytes.

Type: `uint8_t`

Default value: 3

**ITGUMAXBLKS**

Maximum number of ITCM gate unit blocks.  $\text{Number of blocks} = \text{pow}(2, \text{ITGUMAXBLKS})$ .

Type: `uint8_t`

Default value: 0

**ITM**

Level of instrumentation trace supported. `false` : No ITM trace included, `true`: ITM trace included (unless baseline).

Type: `bool`

Default value: `true`

**ITM\_HAS\_LSR**

ITM support LAR and LSR for software lock.

Type: `bool`

Default value: `true`

**LOCKDTGU**

Lock down of Data TGU registers write.

Type: `bool`

Default value: `false`

**LOCKITGU**

Lock down of Instruction TGU registers write.

Type: `bool`

Default value: `false`

**LOCKTCM**

Lock down of TCM registers write.

Type: `bool`

Default value: `false`

### **LOCK\_NS\_MPU**

Lock down of Non-Secure MPU registers write.

Type: `bool`

Default value: `false`

### **LOCK\_SAU**

Lock down of SAU registers write.

Type: `bool`

Default value: `false`

### **LOCK\_S\_MPU**

Lock down of Secure MPU registers write.

Type: `bool`

Default value: `false`

### **LVL\_WIDTH**

Number of bits of interrupt priority (baseline has 2).

Type: `uint8_t`

Default value: 3

### **MEMORY\_REGION\_MASK**

Read/Write Mask for MPU\_RBAR, MPU\_RLAR, SAU\_RBAR, SAU\_RLAR. Bits[4:0] of this parameter are ignored.

Type: `uint32_t`

Default value: `0xffffffff`

### **MPU\_TYPE\_NS.DREGION**

Number of regions in the Non-Secure MPU. If Security Extensions are absent, this is the total number of MPU regions.

Type: `uint16_t`

Default value: 16

**MPU\_TYPE\_S.DREGION**

Number of regions in the Secure MPU. If Security Extensions are absent, this is ignored.

Type: `uint16_t`

Default value: 16

**MVE**

Set whether the model has MVE support. If FPU = 0: 0=MVE not included, 1=Integer subset of MVE included. If FPU = 1: 0=MVE not included, 1=Integer subset of MVE included, 2=Integer and half and single precision floating point MVE included.

Type: `uint8_t`

Default value: 2

**MVFR0.Double-precision**

Support 8-byte floats.

Type: `bool`

Default value: `true`

**MVFR1.FP16**

FP extension implements half-precision floating-point operations. 0 = Not supported; 1 = Supported.

Type: `bool`

Default value: `true`

**MVFR1.FPHP**

FP extension implements half-precision floating-point conversion instructions. 0x1: Half-precision to single-precision, 0x2: As for 0x1 and also half-precision to double-precision.

Type: `uint8_t`

Default value: 2

**MVFR1.MVE**

DEPRECATED: Use parameter MVE instead.

Type: `uint8_t`

Default value: 2

**NUM\_IRQ**

Number of user interrupts.

Type: `uint16_t`

Default value: 16

**NVIC\_ITNS0**

Each character fixes a security state target for a given external interrupt. 'N' implies NS; 'S' implies S; anything else is ignored. The largest bit is first, e.g. 'S-NN' sets external interrupts 0 and 1 to non-secure, 2 remains settable via the NVIC\_ITNS register and 3 always targets secure.

Type: `string`

Default value: N/A

**NVIC\_ITNS1**

Each character fixes a security state target for a given external interrupt. 'N' implies NS; 'S' implies S; anything else is ignored. The largest bit is first, e.g. 'S-NN' sets external interrupts 0 and 1 to non-secure, 2 remains settable via the NVIC\_ITNS register and 3 always targets secure.

Type: `string`

Default value: N/A

**NVIC\_ITNS10**

Each character fixes a security state target for a given external interrupt. 'N' implies NS; 'S' implies S; anything else is ignored. The largest bit is first, e.g. 'S-NN' sets external interrupts 0 and 1 to non-secure, 2 remains settable via the NVIC\_ITNS register and 3 always targets secure.

Type: `string`

Default value: N/A

**NVIC\_ITNS11**

Each character fixes a security state target for a given external interrupt. 'N' implies NS; 'S' implies S; anything else is ignored. The largest bit is first, e.g. 'S-NN' sets external interrupts 0 and 1 to non-secure, 2 remains settable via the NVIC\_ITNS register and 3 always targets secure.

Type: `string`

Default value: N/A

**NVIC\_ITNS12**

Each character fixes a security state target for a given external interrupt. 'N' implies NS; 'S' implies S; anything else is ignored. The largest bit is first, e.g. 'S-NN' sets external interrupts 0 and 1 to non-secure, 2 remains settable via the NVIC\_ITNS register and 3 always targets secure.

Type: `string`

Default value: `N/A`

### **NVIC\_ITNS13**

Each character fixes a security state target for a given external interrupt. 'N' implies NS; 'S' implies S; anything else is ignored. The largest bit is first, e.g. 'S-NN' sets external interrupts 0 and 1 to non-secure, 2 remains settable via the NVIC\_ITNS register and 3 always targets secure.

Type: `string`

Default value: `N/A`

### **NVIC\_ITNS14**

Each character fixes a security state target for a given external interrupt. 'N' implies NS; 'S' implies S; anything else is ignored. The largest bit is first, e.g. 'S-NN' sets external interrupts 0 and 1 to non-secure, 2 remains settable via the NVIC\_ITNS register and 3 always targets secure.

Type: `string`

Default value: `N/A`

### **NVIC\_ITNS15**

Each character fixes a security state target for a given external interrupt. 'N' implies NS; 'S' implies S; anything else is ignored. The largest bit is first, e.g. 'S-NN' sets external interrupts 0 and 1 to non-secure, 2 remains settable via the NVIC\_ITNS register and 3 always targets secure.

Type: `string`

Default value: `N/A`

### **NVIC\_ITNS2**

Each character fixes a security state target for a given external interrupt. 'N' implies NS; 'S' implies S; anything else is ignored. The largest bit is first, e.g. 'S-NN' sets external interrupts 0 and 1 to non-secure, 2 remains settable via the NVIC\_ITNS register and 3 always targets secure.

Type: `string`

Default value: `N/A`

### **NVIC\_ITNS3**

Each character fixes a security state target for a given external interrupt. 'N' implies NS; 'S' implies S; anything else is ignored. The largest bit is first, e.g. 'S-NN' sets external interrupts 0 and 1 to non-secure, 2 remains settable via the NVIC\_ITNS register and 3 always targets secure.

Type: `string`

Default value: `N/A`



**NVIC\_ITNS4**

Each character fixes a security state target for a given external interrupt. 'N' implies NS; 'S' implies S; anything else is ignored. The largest bit is first, e.g. 'S-NN' sets external interrupts 0 and 1 to non-secure, 2 remains settable via the NVIC\_ITNS register and 3 always targets secure.

Type: `string`

Default value: `N/A`

**NVIC\_ITNS5**

Each character fixes a security state target for a given external interrupt. 'N' implies NS; 'S' implies S; anything else is ignored. The largest bit is first, e.g. 'S-NN' sets external interrupts 0 and 1 to non-secure, 2 remains settable via the NVIC\_ITNS register and 3 always targets secure.

Type: `string`

Default value: `N/A`

**NVIC\_ITNS6**

Each character fixes a security state target for a given external interrupt. 'N' implies NS; 'S' implies S; anything else is ignored. The largest bit is first, e.g. 'S-NN' sets external interrupts 0 and 1 to non-secure, 2 remains settable via the NVIC\_ITNS register and 3 always targets secure.

Type: `string`

Default value: `N/A`

**NVIC\_ITNS7**

Each character fixes a security state target for a given external interrupt. 'N' implies NS; 'S' implies S; anything else is ignored. The largest bit is first, e.g. 'S-NN' sets external interrupts 0 and 1 to non-secure, 2 remains settable via the NVIC\_ITNS register and 3 always targets secure.

Type: `string`

Default value: `N/A`

**NVIC\_ITNS8**

Each character fixes a security state target for a given external interrupt. 'N' implies NS; 'S' implies S; anything else is ignored. The largest bit is first, e.g. 'S-NN' sets external interrupts 0 and 1 to non-secure, 2 remains settable via the NVIC\_ITNS register and 3 always targets secure.

Type: `string`

Default value: `N/A`

**NVIC\_ITNS9**

Each character fixes a security state target for a given external interrupt. 'N' implies NS; 'S' implies S; anything else is ignored. The largest bit is first, e.g. 'S-NN' sets external interrupts 0 and 1 to non-secure, 2 remains settable via the NVIC\_ITNS register and 3 always targets secure.

Type: `string`

Default value: `N/A`

**REGISTER\_POP\_ORDER**

Order in which the registers are popped off the stack during exception return. A comma separated list of register names and ranges.

Type: `string`

Default value: `"R4-R11,R0-R3,R12,R14,RETURN_ADDR,CPSR,S0-S15,FPSCR,PADDING,S16-S31"`

**REGISTER\_PUSH\_ORDER**

Order in which the registers are pushed on to the stack during exception handling. A comma separated list of register names and ranges.

Type: `string`

Default value: `"R0-R3,R12,R14,RETURN_ADDR,CPSR,S0-S15,FPSCR,PADDING,S16-S31"`

**SAU\_CTRL.ALLNS**

At reset, the SAU treats entire memory space as NS when the SAU is disabled if this is set.

Type: `bool`

Default value: `false`

**SAU\_CTRL.ENABLE**

Enable SAU at reset.

Type: `bool`

Default value: `false`

**SAU\_TYPE.SREGION**

Number of SAU regions (0 => no SAU).

Type: `uint16_t`

Default value: `16`

**SECEXT**

Whether the ARMv8-M Security Extensions are included.

Type: `bool`

Default value: `true`

**SYST**

Include SysTick timer functionality (0=Absent, 1=Secure only, 2=Secure and NS).

Type: `uint8_t`

Default value: 2

**SYST\_CALIB\_NS\_reset**

SYST\_CALIB\_NS reset value.

Type: `uint32_t`

Default value: 0x0

**SYST\_CALIB\_reset**

SYST\_CALIB reset value.

Type: `uint32_t`

Default value: 0x0

**VTOR\_NS**

NonSecure Vector Table Offset Register is writeable.

Type: `bool`

Default value: `true`

**VTOR\_NS\_MASK**

Non-Secure VTOR write mask.

Type: `uint32_t`

Default value: 0xfffffff80

**VTOR\_S**

Secure Vector Table Offset Register is writeable.

Type: `bool`

Default value: `true`

**VTOR\_S\_MASK**

Secure VTOR write mask.

Type: `uint32_t`

Default value: `0xffffffff80`

**WIC**

Include support for WIC-mode deep sleep.

Type: `bool`

Default value: `true`

**abort\_unaligned\_nonNormal**

If true, **UNPREDICTABLE** accesses of device and strongly ordered memory abort; if false they are allowed.

Type: `bool`

Default value: `true`

**aircr\_iesb\_is\_writable**

IS the AIRCR.IESB bit [5] writable?.

Type: `bool`

Default value: `true`

**aircr\_iesb\_reset**

Set the AIRCR.IESB bit [5] after reset.

Type: `bool`

Default value: `false`

**allow\_dap\_writes\_while\_core\_running**

Debug writes are respected even while the core is running, i.e. the core does not have to be halted.

Type: `bool`

Default value: `true`

**allow\_debug\_monitor\_with\_in\_flight\_inst**

Allow handling Debug Monitor exception with in-flight instructions.

Type: `bool`

Default value: `false`

**allow\_stack\_accesses\_to\_ppb\_space**

Allow stack accesses to PPB space.

Type: `bool`

Default value: `false`

**always\_undefinstr\_over\_nocp**

Only v8.0M. Always fault with UNDEFINSTR for undefined instructions that fall in CP space (don't check coprocessor status).

Type: `bool`

Default value: `false`

**apply\_prigroup\_to\_pending\_tree**

DEPRECATED: please use `sep_sec_state_then_apply_prigroup_to_pending`. Original description: Apply AIRCR.PRIGROUP to the pending and active trees (instead of just the active tree) when selecting the highest priority pending exception.

Type: `bool`

Default value: `false`

**baseline**

Use the baseline profile (if false, use mainline).

Type: `bool`

Default value: `true`

**bp\_on\_2nd\_halfword**

Respect DWT/BPU breakpoint-hit on 2nd halfword of 32-bit instruction.

Type: `bool`

Default value: `true`

**callee\_register\_push\_low\_to\_high**

If true, push callee registers in order from R4 to R11. If false, push R11 to R4.

Type: `bool`

Default value: `true`

**cde\_fp\_check\_on\_unsupported**

Run FP checks on both supported and unsupported CDE instructions.

Type: `bool`

Default value: `false`

**cde\_impl\_name**

Name of the CDE implementation for this core (implementation contributed by MTI plugin).

Type: `string`

Default value: `N/A`

**clear\_non\_secure\_EXC\_RETURN.ES\_on\_tailchain**

Clear EXC\_RETURN.ES in LR value on entry to a tail-chained exception when returning from Non-secure state.

Type: `bool`

Default value: `true`

**condition\_flags\_reset**

Reset Value of condition flags in APSR.

Type: `uint8_t`

Default value: `0`

**cpi\_div**

divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**cpi\_mul**

multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**cpu\_can\_access\_debug\_regs**

The DWT, BPU, ROM table, DCB, and the SHCSR and DFSR registers access from the processor.

Type: `bool`

Default value: `true`

### **dbg\_coproc\_load\_store\_enable**

Enable LDCX and STCX instructions.

Type: `uint8_t`

Default value: `0`

### **dcache-invalidate-ns-cleans-s**

Whether V8M DCI\* in non-secure should clean-and-invalidate secure cache contents.

Type: `bool`

Default value: `false`

### **dcache-size**

L1 D-cache size in bytes.

Type: `uint32_t`

Default value: `0x8000`

### **dcache-state\_modelled**

Set whether D-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **delay\_faultmask\_update**

Delay FAULTMASK update to context sync.

Type: `bool`

Default value: `false`

### **delay\_sysreg\_update**

Delay some system register updates (e.g. SHCSR) to context sync.

Type: `bool`

Default value: `false`

### **do\_exclusive\_monitor\_check\_first**

In exclusive stores, check local exclusive monitor before detecting other memory aborts.

Type: `bool`

Default value: `false`

**drop\_mem\_fault**

Whether to drop `mem_fault` in favour of subsequently generated NOCP/secure fault in PushStack.

Type: `bool`

Default value: `false`

**dtcm\_enable**

Enable DTCM at reset.

Type: `bool`

Default value: `false`

**dtcm\_size**

DTCM size in KB.

Type: `uint16_t`

Default value: `0x100`

**dwt\_unaligned\_word\_access\_as\_half\_word**

DWT Treat unaligned word access as half word or bytes.

Type: `bool`

Default value: `true`

**enable\_helium\_extension**

Enable Helium extension.

Type: `bool`

Default value: `false`

**exercise\_strex\_fail**

Reject a pseudo-random majority of exclusive store instructions.

Type: `bool`

Default value: `false`

**has\_ahbp**

Are Vendor-Sys accesses sent to a separate bus (AHBP on CM7).

Type: `bool`



Default value: `true`

**has\_arm\_v8-1m**

Enable v8.1M architecture version and features.

Type: `bool`

Default value: `false`

**has\_cde**

Enables Custom Datapath Extensions.

Type: `bool`

Default value: `false`

**has\_core\_dside\_bus\_gasket**

STL gasket enabled.

Type: `bool`

Default value: `false`

**has\_lob\_cache**

Support for LOB cache (only if support for LO instructions is enabled as well).

Type: `bool`

Default value: `true`

**has\_m55\_tcmcr**

If true, enables the CortexM55 TCM Control Registers (ITCMCR and DTCMCR), If false, CortexM55 TCM Control Registers are disabled.

Type: `bool`

Default value: `false`

**has\_pmu**

Availability of optional PMU.

Type: `bool`

Default value: `false`

**has\_separate\_etm\_reset**

If true, signal 'etmreset' resets the core, else the core power-on-reset does.

Type: `bool`

Default value: `false`

### **has\_unpriviledged\_debug**

Unprivileged Debug Extension supported for Mainline Extension.

Type: `bool`

Default value: `true`

### **has\_writebuffer**

Implement write accesses buffering before L1 cache. May affect `ext_abort` behaviour.

Type: `bool`

Default value: `false`

### **icache-size**

L1 I-cache size in bytes.

Type: `uint32_t`

Default value: `0x8000`

### **icache-state\_modelled**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **ignore\_RNR\_top\_nibble**

If set, only the bottom four bits of `MPU_RNR.REGION` are used.

Type: `bool`

Default value: `false`

### **ignore\_demcr\_sdme\_for\_nonhalting\_bkpt**

Ignore the SDME bit of the DEMCR register when escalating a Debug Monitor exception to a HardFault.

Type: `bool`

Default value: `false`

**ignore\_out\_of\_range\_RNR\_write**

If an MPU\_RNR.REGION write is out of range, ignore it ; if false, MPU\_RNR values wrap.

Type: `bool`

Default value: `false`

**ignore\_unpred\_SBZSBO**

Use smaller decoder does not UNDEF some unpredictable SBZ/SBO fields.

Type: `bool`

Default value: `false`

**ignore\_unpred\_ZeroRegistersInList**

VLDM,VSTM,STM,LDM with no registers **NOP** instead of UNDEF.

Type: `bool`

Default value: `false`

**itcm\_enable**

Enable ITCM at reset.

Type: `bool`

Default value: `false`

**itcm\_size**

ITCM size in KB.

Type: `uint16_t`

Default value: `0x100`

**late\_arrival**

Enable late arrival support.

Type: `bool`

Default value: `true`

**manager\_id**

Manager ID presented in bus transactions.

Type: `uint64_t`

Default value: `0x0`

**min\_sync\_level**

force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: 0

**mve\_has\_atomic\_ticks**

Enable atomic ticks behaviour for vector instructions flagged as such (e.g. VLDR).

Type: `bool`

Default value: `false`

**num\_pmu\_counters**

Number of available PMU counters.

Type: `uint8_t`

Default value: 31

**number\_of\_itm\_stimulus\_ports**

The number of ITM stimulus ports.

Type: `uint16_t`

Default value: 32

**pend\_overriden\_exception\_on\_stack\_push**

Mark any overridden exceptions on stack push as pending (instead of dropping them).

Type: `bool`

Default value: `false`

**preserve\_unknown\_caller\_save\_regs\_at\_S\_to\_S**

preserve unknown caller registers when they become **UNKNOWN** at secure to secure.

Type: `bool`

Default value: `true`

**ras\_ERRFR0**

A JSON object or array of objects for each field of ERRFR. Records not described default to **RAZ** e.g. `{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}`.

Type: string

Default value: "'{'ED':0x2, 'UI':0x2, 'FI':0x2, 'UE':0x2, 'CFI':0x2, 'CEC':0x2, 'RP':0x1, 'DUI':0x0, 'CEO':0x1, 'CI':0x2}'"

### **ras\_ERRFR1**

A JSON object or array of objects for each field of ERRFR. Records not described default to **RAZ** e.g. {'ED':0x2, 'UI':0x2, 'FI':0x2, 'UE':0x2, 'CFI':0x2, 'CEC':0x2, 'RP':0x1, 'DUI':0x0, 'CEO':0x1, 'CI':0x2}'.

Type: string

Default value: "'{'ED':0x2, 'UI':0x2, 'FI':0x2, 'UE':0x2, 'CFI':0x2, 'CEC':0x2, 'RP':0x1, 'DUI':0x0, 'CEO':0x1, 'CI':0x2}'"

### **ras\_ERRFR10**

A JSON object or array of objects for each field of ERRFR. Records not described default to **RAZ** e.g. {'ED':0x2, 'UI':0x2, 'FI':0x2, 'UE':0x2, 'CFI':0x2, 'CEC':0x2, 'RP':0x1, 'DUI':0x0, 'CEO':0x1, 'CI':0x2}'.

Type: string

Default value: "'{'ED':0x2, 'UI':0x2, 'FI':0x2, 'UE':0x2, 'CFI':0x2, 'CEC':0x2, 'RP':0x1, 'DUI':0x0, 'CEO':0x1, 'CI':0x2}'"

### **ras\_ERRFR11**

A JSON object or array of objects for each field of ERRFR. Records not described default to **RAZ** e.g. {'ED':0x2, 'UI':0x2, 'FI':0x2, 'UE':0x2, 'CFI':0x2, 'CEC':0x2, 'RP':0x1, 'DUI':0x0, 'CEO':0x1, 'CI':0x2}'.

Type: string

Default value: "'{'ED':0x2, 'UI':0x2, 'FI':0x2, 'UE':0x2, 'CFI':0x2, 'CEC':0x2, 'RP':0x1, 'DUI':0x0, 'CEO':0x1, 'CI':0x2}'"

### **ras\_ERRFR12**

A JSON object or array of objects for each field of ERRFR. Records not described default to **RAZ** e.g. {'ED':0x2, 'UI':0x2, 'FI':0x2, 'UE':0x2, 'CFI':0x2, 'CEC':0x2, 'RP':0x1, 'DUI':0x0, 'CEO':0x1, 'CI':0x2}'.

Type: string

Default value: "'{'ED':0x2, 'UI':0x2, 'FI':0x2, 'UE':0x2, 'CFI':0x2, 'CEC':0x2, 'RP':0x1, 'DUI':0x0, 'CEO':0x1, 'CI':0x2}'"

**ras\_ERRFR13**

A JSON object or array of objects for each field of ERRFR. Records not described default to **RAZ** e.g. {"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}.

Type: string

Default value: "{ \"ED\":0x2, \"UI\":0x2, \"FI\":0x2, \"UE\":0x2, \"CFI\":0x2, \"CEC\":0x2, \"RP\":0x1, \"DUI\":0x0, \"CEO\":0x1, \"CI\":0x2} "

**ras\_ERRFR14**

A JSON object or array of objects for each field of ERRFR. Records not described default to **RAZ** e.g. {"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}.

Type: string

Default value: "{ \"ED\":0x2, \"UI\":0x2, \"FI\":0x2, \"UE\":0x2, \"CFI\":0x2, \"CEC\":0x2, \"RP\":0x1, \"DUI\":0x0, \"CEO\":0x1, \"CI\":0x2} "

**ras\_ERRFR15**

A JSON object or array of objects for each field of ERRFR. Records not described default to **RAZ** e.g. {"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}.

Type: string

Default value: "{ \"ED\":0x2, \"UI\":0x2, \"FI\":0x2, \"UE\":0x2, \"CFI\":0x2, \"CEC\":0x2, \"RP\":0x1, \"DUI\":0x0, \"CEO\":0x1, \"CI\":0x2} "

**ras\_ERRFR16**

A JSON object or array of objects for each field of ERRFR. Records not described default to **RAZ** e.g. {"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}.

Type: string

Default value: "{ \"ED\":0x2, \"UI\":0x2, \"FI\":0x2, \"UE\":0x2, \"CFI\":0x2, \"CEC\":0x2, \"RP\":0x1, \"DUI\":0x0, \"CEO\":0x1, \"CI\":0x2} "

**ras\_ERRFR17**

A JSON object or array of objects for each field of ERRFR. Records not described default to **RAZ** e.g. {"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}.

Type: string

Default value: "'{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'"

### **ras\_ERRFR18**

A JSON object or array of objects for each field of ERRFR. Records not described default to **RAZ** e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

Type: string

Default value: "'{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'"

### **ras\_ERRFR19**

A JSON object or array of objects for each field of ERRFR. Records not described default to **RAZ** e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

Type: string

Default value: "'{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'"

### **ras\_ERRFR2**

A JSON object or array of objects for each field of ERRFR. Records not described default to **RAZ** e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

Type: string

Default value: "'{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'"

### **ras\_ERRFR20**

A JSON object or array of objects for each field of ERRFR. Records not described default to **RAZ** e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

Type: string

Default value: "'{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'"

### **ras\_ERRFR21**

A JSON object or array of objects for each field of ERRFR. Records not described default to **RAZ** e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

Type: string

Default value: "'{'\"ED\":0x2, \"UI\":0x2, \"FI\":0x2, \"UE\":0x2, \"CFI\":0x2, \"CEC\":0x2, \"RP\":0x1, \"DUI\":0x0, \"CEO\":0x1, \"CI\":0x2}'"

### **ras\_ERRFR22**

A JSON object or array of objects for each field of ERRFR. Records not described default to **RAZ** e.g. {'ED':0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

Type: string

Default value: "'{'\"ED\":0x2, \"UI\":0x2, \"FI\":0x2, \"UE\":0x2, \"CFI\":0x2, \"CEC\":0x2, \"RP\":0x1, \"DUI\":0x0, \"CEO\":0x1, \"CI\":0x2}'"

### **ras\_ERRFR23**

A JSON object or array of objects for each field of ERRFR. Records not described default to **RAZ** e.g. {'ED':0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

Type: string

Default value: "'{'\"ED\":0x2, \"UI\":0x2, \"FI\":0x2, \"UE\":0x2, \"CFI\":0x2, \"CEC\":0x2, \"RP\":0x1, \"DUI\":0x0, \"CEO\":0x1, \"CI\":0x2}'"

### **ras\_ERRFR24**

A JSON object or array of objects for each field of ERRFR. Records not described default to **RAZ** e.g. {'ED':0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

Type: string

Default value: "'{'\"ED\":0x2, \"UI\":0x2, \"FI\":0x2, \"UE\":0x2, \"CFI\":0x2, \"CEC\":0x2, \"RP\":0x1, \"DUI\":0x0, \"CEO\":0x1, \"CI\":0x2}'"

### **ras\_ERRFR25**

A JSON object or array of objects for each field of ERRFR. Records not described default to **RAZ** e.g. {'ED':0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

Type: string

Default value: "'{'\"ED\":0x2, \"UI\":0x2, \"FI\":0x2, \"UE\":0x2, \"CFI\":0x2, \"CEC\":0x2, \"RP\":0x1, \"DUI\":0x0, \"CEO\":0x1, \"CI\":0x2}'"



**ras\_ERRFR26**

A JSON object or array of objects for each field of ERRFR. Records not described default to **RAZ** e.g. {"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}.

Type: string

Default value: "{ \"ED\":0x2, \"UI\":0x2, \"FI\":0x2, \"UE\":0x2, \"CFI\":0x2, \"CEC\":0x2, \"RP\":0x1, \"DUI\":0x0, \"CEO\":0x1, \"CI\":0x2} "

**ras\_ERRFR27**

A JSON object or array of objects for each field of ERRFR. Records not described default to **RAZ** e.g. {"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}.

Type: string

Default value: "{ \"ED\":0x2, \"UI\":0x2, \"FI\":0x2, \"UE\":0x2, \"CFI\":0x2, \"CEC\":0x2, \"RP\":0x1, \"DUI\":0x0, \"CEO\":0x1, \"CI\":0x2} "

**ras\_ERRFR28**

A JSON object or array of objects for each field of ERRFR. Records not described default to **RAZ** e.g. {"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}.

Type: string

Default value: "{ \"ED\":0x2, \"UI\":0x2, \"FI\":0x2, \"UE\":0x2, \"CFI\":0x2, \"CEC\":0x2, \"RP\":0x1, \"DUI\":0x0, \"CEO\":0x1, \"CI\":0x2} "

**ras\_ERRFR29**

A JSON object or array of objects for each field of ERRFR. Records not described default to **RAZ** e.g. {"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}.

Type: string

Default value: "{ \"ED\":0x2, \"UI\":0x2, \"FI\":0x2, \"UE\":0x2, \"CFI\":0x2, \"CEC\":0x2, \"RP\":0x1, \"DUI\":0x0, \"CEO\":0x1, \"CI\":0x2} "

**ras\_ERRFR3**

A JSON object or array of objects for each field of ERRFR. Records not described default to **RAZ** e.g. {"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}.

Type: string

Default value: "'{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'"

### **ras\_ERRFR30**

A JSON object or array of objects for each field of ERRFR. Records not described default to **RAZ** e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

Type: string

Default value: "'{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'"

### **ras\_ERRFR31**

A JSON object or array of objects for each field of ERRFR. Records not described default to **RAZ** e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

Type: string

Default value: "'{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'"

### **ras\_ERRFR32**

A JSON object or array of objects for each field of ERRFR. Records not described default to **RAZ** e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

Type: string

Default value: "'{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'"

### **ras\_ERRFR33**

A JSON object or array of objects for each field of ERRFR. Records not described default to **RAZ** e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

Type: string

Default value: "'{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'"

### **ras\_ERRFR34**

A JSON object or array of objects for each field of ERRFR. Records not described default to **RAZ** e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

Type: string

Default value: "'{'\"ED\":0x2, \"UI\":0x2, \"FI\":0x2, \"UE\":0x2, \"CFI\":0x2, \"CEC\":0x2, \"RP\":0x1, \"DUI\":0x0, \"CEO\":0x1, \"CI\":0x2}'"

### **ras\_ERRFR35**

A JSON object or array of objects for each field of ERRFR. Records not described default to **RAZ** e.g. {'\"ED\":0x2, \"UI\":0x2, \"FI\":0x2, \"UE\":0x2, \"CFI\":0x2, \"CEC\":0x2, \"RP\":0x1, \"DUI\":0x0, \"CEO\":0x1, \"CI\":0x2}'.

Type: string

Default value: "'{'\"ED\":0x2, \"UI\":0x2, \"FI\":0x2, \"UE\":0x2, \"CFI\":0x2, \"CEC\":0x2, \"RP\":0x1, \"DUI\":0x0, \"CEO\":0x1, \"CI\":0x2}'"

### **ras\_ERRFR36**

A JSON object or array of objects for each field of ERRFR. Records not described default to **RAZ** e.g. {'\"ED\":0x2, \"UI\":0x2, \"FI\":0x2, \"UE\":0x2, \"CFI\":0x2, \"CEC\":0x2, \"RP\":0x1, \"DUI\":0x0, \"CEO\":0x1, \"CI\":0x2}'.

Type: string

Default value: "'{'\"ED\":0x2, \"UI\":0x2, \"FI\":0x2, \"UE\":0x2, \"CFI\":0x2, \"CEC\":0x2, \"RP\":0x1, \"DUI\":0x0, \"CEO\":0x1, \"CI\":0x2}'"

### **ras\_ERRFR37**

A JSON object or array of objects for each field of ERRFR. Records not described default to **RAZ** e.g. {'\"ED\":0x2, \"UI\":0x2, \"FI\":0x2, \"UE\":0x2, \"CFI\":0x2, \"CEC\":0x2, \"RP\":0x1, \"DUI\":0x0, \"CEO\":0x1, \"CI\":0x2}'.

Type: string

Default value: "'{'\"ED\":0x2, \"UI\":0x2, \"FI\":0x2, \"UE\":0x2, \"CFI\":0x2, \"CEC\":0x2, \"RP\":0x1, \"DUI\":0x0, \"CEO\":0x1, \"CI\":0x2}'"

### **ras\_ERRFR38**

A JSON object or array of objects for each field of ERRFR. Records not described default to **RAZ** e.g. {'\"ED\":0x2, \"UI\":0x2, \"FI\":0x2, \"UE\":0x2, \"CFI\":0x2, \"CEC\":0x2, \"RP\":0x1, \"DUI\":0x0, \"CEO\":0x1, \"CI\":0x2}'.

Type: string

Default value: "'{'\"ED\":0x2, \"UI\":0x2, \"FI\":0x2, \"UE\":0x2, \"CFI\":0x2, \"CEC\":0x2, \"RP\":0x1, \"DUI\":0x0, \"CEO\":0x1, \"CI\":0x2}'"

**ras\_ERRFR39**

A JSON object or array of objects for each field of ERRFR. Records not described default to **RAZ** e.g. {"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}.

Type: string

Default value: "{ \"ED\":0x2, \"UI\":0x2, \"FI\":0x2, \"UE\":0x2, \"CFI\":0x2, \"CEC\":0x2, \"RP\":0x1, \"DUI\":0x0, \"CEO\":0x1, \"CI\":0x2} "

**ras\_ERRFR4**

A JSON object or array of objects for each field of ERRFR. Records not described default to **RAZ** e.g. {"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}.

Type: string

Default value: "{ \"ED\":0x2, \"UI\":0x2, \"FI\":0x2, \"UE\":0x2, \"CFI\":0x2, \"CEC\":0x2, \"RP\":0x1, \"DUI\":0x0, \"CEO\":0x1, \"CI\":0x2} "

**ras\_ERRFR40**

A JSON object or array of objects for each field of ERRFR. Records not described default to **RAZ** e.g. {"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}.

Type: string

Default value: "{ \"ED\":0x2, \"UI\":0x2, \"FI\":0x2, \"UE\":0x2, \"CFI\":0x2, \"CEC\":0x2, \"RP\":0x1, \"DUI\":0x0, \"CEO\":0x1, \"CI\":0x2} "

**ras\_ERRFR41**

A JSON object or array of objects for each field of ERRFR. Records not described default to **RAZ** e.g. {"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}.

Type: string

Default value: "{ \"ED\":0x2, \"UI\":0x2, \"FI\":0x2, \"UE\":0x2, \"CFI\":0x2, \"CEC\":0x2, \"RP\":0x1, \"DUI\":0x0, \"CEO\":0x1, \"CI\":0x2} "

**ras\_ERRFR42**

A JSON object or array of objects for each field of ERRFR. Records not described default to **RAZ** e.g. {"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}.

Type: string

Default value: "'{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'"

### **ras\_ERRFR43**

A JSON object or array of objects for each field of ERRFR. Records not described default to **RAZ** e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

Type: string

Default value: "'{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'"

### **ras\_ERRFR44**

A JSON object or array of objects for each field of ERRFR. Records not described default to **RAZ** e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

Type: string

Default value: "'{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'"

### **ras\_ERRFR45**

A JSON object or array of objects for each field of ERRFR. Records not described default to **RAZ** e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

Type: string

Default value: "'{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'"

### **ras\_ERRFR46**

A JSON object or array of objects for each field of ERRFR. Records not described default to **RAZ** e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

Type: string

Default value: "'{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'"

### **ras\_ERRFR47**

A JSON object or array of objects for each field of ERRFR. Records not described default to **RAZ** e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

Type: string

Default value: `"{'\ED\":0x2, \"UI\":0x2, \"FI\":0x2, \"UE\":0x2, \"CFI\":0x2, \"CEC\":0x2, \"RP\":0x1, \"DUI\":0x0, \"CEO\":0x1, \"CI\":0x2}'"`

#### **ras\_ERRFR48**

A JSON object or array of objects for each field of ERRFR. Records not described default to **RAZ** e.g. `{\"ED\":0x2, \"UI\":0x2, \"FI\":0x2, \"UE\":0x2, \"CFI\":0x2, \"CEC\":0x2, \"RP\":0x1, \"DUI\":0x0, \"CEO\":0x1, \"CI\":0x2}'`.

Type: string

Default value: `"{'\ED\":0x2, \"UI\":0x2, \"FI\":0x2, \"UE\":0x2, \"CFI\":0x2, \"CEC\":0x2, \"RP\":0x1, \"DUI\":0x0, \"CEO\":0x1, \"CI\":0x2}'"`

#### **ras\_ERRFR49**

A JSON object or array of objects for each field of ERRFR. Records not described default to **RAZ** e.g. `{\"ED\":0x2, \"UI\":0x2, \"FI\":0x2, \"UE\":0x2, \"CFI\":0x2, \"CEC\":0x2, \"RP\":0x1, \"DUI\":0x0, \"CEO\":0x1, \"CI\":0x2}'`.

Type: string

Default value: `"{'\ED\":0x2, \"UI\":0x2, \"FI\":0x2, \"UE\":0x2, \"CFI\":0x2, \"CEC\":0x2, \"RP\":0x1, \"DUI\":0x0, \"CEO\":0x1, \"CI\":0x2}'"`

#### **ras\_ERRFR5**

A JSON object or array of objects for each field of ERRFR. Records not described default to **RAZ** e.g. `{\"ED\":0x2, \"UI\":0x2, \"FI\":0x2, \"UE\":0x2, \"CFI\":0x2, \"CEC\":0x2, \"RP\":0x1, \"DUI\":0x0, \"CEO\":0x1, \"CI\":0x2}'`.

Type: string

Default value: `"{'\ED\":0x2, \"UI\":0x2, \"FI\":0x2, \"UE\":0x2, \"CFI\":0x2, \"CEC\":0x2, \"RP\":0x1, \"DUI\":0x0, \"CEO\":0x1, \"CI\":0x2}'"`

#### **ras\_ERRFR50**

A JSON object or array of objects for each field of ERRFR. Records not described default to **RAZ** e.g. `{\"ED\":0x2, \"UI\":0x2, \"FI\":0x2, \"UE\":0x2, \"CFI\":0x2, \"CEC\":0x2, \"RP\":0x1, \"DUI\":0x0, \"CEO\":0x1, \"CI\":0x2}'`.

Type: string

Default value: `"{'\ED\":0x2, \"UI\":0x2, \"FI\":0x2, \"UE\":0x2, \"CFI\":0x2, \"CEC\":0x2, \"RP\":0x1, \"DUI\":0x0, \"CEO\":0x1, \"CI\":0x2}'"`

**ras\_ERRFR51**

A JSON object or array of objects for each field of ERRFR. Records not described default to **RAZ** e.g. {"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}.

Type: string

Default value: "{ \"ED\":0x2, \"UI\":0x2, \"FI\":0x2, \"UE\":0x2, \"CFI\":0x2, \"CEC\":0x2, \"RP\":0x1, \"DUI\":0x0, \"CEO\":0x1, \"CI\":0x2} \"

**ras\_ERRFR52**

A JSON object or array of objects for each field of ERRFR. Records not described default to **RAZ** e.g. {"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}.

Type: string

Default value: "{ \"ED\":0x2, \"UI\":0x2, \"FI\":0x2, \"UE\":0x2, \"CFI\":0x2, \"CEC\":0x2, \"RP\":0x1, \"DUI\":0x0, \"CEO\":0x1, \"CI\":0x2} \"

**ras\_ERRFR53**

A JSON object or array of objects for each field of ERRFR. Records not described default to **RAZ** e.g. {"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}.

Type: string

Default value: "{ \"ED\":0x2, \"UI\":0x2, \"FI\":0x2, \"UE\":0x2, \"CFI\":0x2, \"CEC\":0x2, \"RP\":0x1, \"DUI\":0x0, \"CEO\":0x1, \"CI\":0x2} \"

**ras\_ERRFR54**

A JSON object or array of objects for each field of ERRFR. Records not described default to **RAZ** e.g. {"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}.

Type: string

Default value: "{ \"ED\":0x2, \"UI\":0x2, \"FI\":0x2, \"UE\":0x2, \"CFI\":0x2, \"CEC\":0x2, \"RP\":0x1, \"DUI\":0x0, \"CEO\":0x1, \"CI\":0x2} \"

**ras\_ERRFR55**

A JSON object or array of objects for each field of ERRFR. Records not described default to **RAZ** e.g. {"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}.

Type: string

Default value: "'{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'"

### **ras\_ERRFR6**

A JSON object or array of objects for each field of ERRFR. Records not described default to **RAZ** e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

Type: string

Default value: "'{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'"

### **ras\_ERRFR7**

A JSON object or array of objects for each field of ERRFR. Records not described default to **RAZ** e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

Type: string

Default value: "'{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'"

### **ras\_ERRFR8**

A JSON object or array of objects for each field of ERRFR. Records not described default to **RAZ** e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

Type: string

Default value: "'{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'"

### **ras\_ERRFR9**

A JSON object or array of objects for each field of ERRFR. Records not described default to **RAZ** e.g. '{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'.

Type: string

Default value: "'{"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x2, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x1, "CI":0x2}'"

### **ras\_cei\_pin**

RAS: Critical error interrupt pin.

Type: uint16\_t



Default value: 2

**ras\_cei\_support**

RAS: Whether Critical Error Interrupt is supported.

Type: bool

Default value: true

**ras\_eri\_pin**

RAS: Error recovery interrupt pin.

Type: uint16\_t

Default value: 1

**ras\_eri\_support**

RAS: Whether Error Recovery Interrupt is supported.

Type: bool

Default value: true

**ras\_error\_record**

56 bit value that specifies which nodes out of 0-55 are implemented (ERRDEVID is derived from this parameter).

Type: uint64\_t

Default value: 0xffffffffffffffff

**ras\_fhi\_pin**

RAS: Fault handling interrupt pin.

Type: uint16\_t

Default value: 0

**ras\_fhi\_support**

RAS: Whether Fault Handling Interrupt is supported.

Type: bool

Default value: true

**rd\_ns\_bus\_err\_behave**

External read aborts in nonsecure domain 0:ignored, 1:precise, 2:imprecise, 3=imprecise except SO.

Type: `uint8_t`

Default value: 1

### **`rd_s_bus_err_behave`**

External read aborts in secure domain 0:ignored, 1:precise, 2:imprecise, 3=imprecise except SO.

Type: `uint8_t`

Default value: 1

### **`register_reset_data`**

Data used to fill register bits when they become **UNKNOWN** at reset.

Type: `uint32_t`

Default value: 0x0

### **`reported_patch_level`**

Purely cosmetic patch level value to be displayed from the CPUID register. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: `int8_t`

Default value: -1

### **`reported_revision_number`**

Purely cosmetic revision number value to be displayed from the CPUID register. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: `int8_t`

Default value: -1

### **`semihosting-Thumb_SVC`**

T32 SVC number for semihosting.

Type: `uint8_t`

Default value: 171

### **`semihosting-cmd_line`**

Command line available to semihosting SVC calls.

Type: `string`

Default value: N/A

**semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**semihosting-enable**

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

Type: `bool`

Default value: `true`

**semihosting-heap\_base**

Virtual address of heap base.

Type: `uint32_t`

Default value: `0x0`

**semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint32_t`

Default value: `0x20700000`

**semihosting-prefix**

Prefix semihosting output with target instance name.

Type: `bool`

Default value: `false`

**semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint32_t`

Default value: `0x20800000`

**semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint32_t`

Default value: 0x20700000

### **sep\_sec\_state\_then\_apply\_prigroup\_to\_pending**

Use separate comparison trees for Secure and Non-Secure pending exceptions and apply AIRCR.PRIGROUP to the output of each before they are compared to determine the overall highest priority.

Type: `bool`

Default value: `false`

### **sequential\_security\_transitions**

Allow transition of security state in sequential instruction fetches that cross from non-secure to secure memory with SG instruction 0: never, 1: always, 2: 32-bit instrs, 3: ISB.

Type: `uint8_t`

Default value: 1

### **share\_fault\_address\_reg**

If true, Fault Address Register is shared.

Type: `bool`

Default value: `false`

### **stack\_limit\_check**

Implementation defined stack limit checks for instructions. Bit 0: Load-exclusive, Bit 1: Load-acquire, Bit 2: VLDM. Any instruction that can't be configured does stack limit check by default.

Type: `uint8_t`

Default value: 7

### **stack\_limit\_check\_optimization**

Stack limit check optimization (0: limit check done for each word on the stack, 1: limit check done only on stack pointer).

Type: `bool`

Default value: `true`

### **stacking\_writes\_are\_precise**

Faults on stack writes are precise.

Type: `bool`

Default value: `true`

**supports\_unprivileged**

Enable support for Unprivileged/Privileged Extension.

Type: `bool`

Default value: `true`

**tail\_chain**

Enable tail-chaining optimisation.

Type: `bool`

Default value: `true`

**trace\_style**

MVE instruction trace style: Add 16 for `[*-]` beat trace. Add 32 for tracing IMPLIED LOB instructions. Add 64 to change opcode of implied BF to `0xBF00`.

Type: `uint8_t`

Default value: `2`

**unknown\_regs\_at\_exception\_value**

Data used to fill registers when they become **UNKNOWN** at exception and exception-return.

Type: `uint32_t`

Default value: `0x0`

**unpred\_WriteBackandBaseInList\_stores\_old\_base\_value**

allow STM with write back to base register in register list.

Type: `bool`

Default value: `false`

**unpred\_mon\_step\_write**

Behavior on unpredictable updates to MON\_STEP bit of DEMCR. 0: ignore write, 1: set one, 2: set zero.

Type: `uint8_t`

Default value: `0`

**unpred\_msr\_psr\_with\_one\_mask\_and\_nodsp\_is\_nop**

If true, MSR to \*PSR with a one mask and no DSP does nothing.

Type: `bool`

Default value: `true`

#### **`unpred_msr_psr_with_zero_mask_is_nop`**

If true, MSR to \*PSR with a zero mask does nothing.

Type: `bool`

Default value: `false`

#### **`unstack_R_regs_before_fp_cp_check`**

In exception return unstack normal register before checking fp coprocessor is enable to unstack FP register.

Type: `bool`

Default value: `false`

#### **`vector_fetch_as_wpt_event`**

Watchpoint on exception vector fetch.

Type: `bool`

Default value: `false`

#### **`vector_fetch_busfault_sets_HFSR_FORCED`**

Only v8.0M. Set HFSR.FORCED when a vector table read generates a HardFault.

Type: `bool`

Default value: `false`

#### **`vector_fetch_on_iside`**

Perform vector fetch on I-side.

Type: `bool`

Default value: `true`

#### **`vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**vfp-present**

Set whether the model has VFP support.

Type: `bool`

Default value: `true`

**wr\_ns\_bus\_err\_behave**

External write aborts in nonsecure domain 0: ignored, 1: precise, 2: imprecise, 3=imprecise except SO.

Type: `uint8_t`

Default value: 3

**wr\_s\_bus\_err\_behave**

External write aborts in secure domain 0: ignored, 1: precise, 2: imprecise, 3=imprecise except SO.

Type: `uint8_t`

Default value: 3

**write\_unknown\_regs\_at\_exception**

Do we write registers when they become **UNKNOWN** at exception or exception-return.

Type: `bool`

Default value: `false`

## 3.32 ARMC1NanoCT

Defined in `LISA/ARMC1NanoCT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Preliminary support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following parameters were added:

- `has_delayed_dbgreg`
- `has_delayed_sysreg`
- `num_acp`

## About ARMC1NanoCT

The following example platforms are available:

- FVP\_Base\_C1-Nano
- EVS\_Dhrystone\_C1-Nanox1
- SVP\_Base\_C1-Nanox1

The following functionality is supported in this release:

- C1-SME2 is supported with the following limitations:
  - The arbitration and assignment logic is not supported.
  - RAS error handling is supported in the first CME only.
  - The C1-SME2 auxiliary AMU counters are not supported.
  - It is assumed that C1-SME2 units are operating in dynamic mode, with a minimum power mode of OFF. Accompanying PPU registers may be read or written to, but they do not determine the behavior of the CME. The PPU interrupt signal for the C1-SME2 is also not modeled or exposed at this time.
- Pilatus DSU support for C1-SME2.
- L2Cache is supported at the per-core level only and there is no implementation of Core-Complex with shared L2Cache yet.
- BROADCASTPERSIST pin is implemented.
- Optional peripheral port is supported.
- L3Cache partition is supported.
- Per-core clock is supported.

## Limitations

The following features are not yet supported, and will be added in a future release:

- No support for Core-Complex yet.
- BROADCASTCACHEMAINTPOU pin is not implemented.
- COREINSTRRET signals are not implemented.
- MEM\_RET power mode is not supported.

The following features are not implemented:

- DynamIQ features that are negligible to the programmers' view simulation will not be implemented in the Fast Model.
- 256-bit wide output transactions will not be supported.
- Error correction/detection features will not be supported.
- Self-test features (MBIST) will not be supported.
- Latency configuration will not be supported.
- Snoop filtering will not be supported.



- Cache stashing capability will not be supported.

## Iris and MTI instances for ARMC1NanoCT

This model has the following Iris instances:

Name	Instance type
ARMC1NanoCT	Cluster_ARM_C1-Nano
ARMC1NanoCT.AMU	PVBusLogger
ARMC1NanoCT.AMU.mapper	PVBusMapper
ARMC1NanoCT.DAP	PVBusLogger
ARMC1NanoCT.DAP.mapper	PVBusMapper
ARMC1NanoCT.DSU	C1-DSU
ARMC1NanoCT.DSU.PPU_cluster	PPUv1
ARMC1NanoCT.DSU.PPU_cluster.busslave	PVBusSlave
ARMC1NanoCT.DSU.PPU_core0	PPUv1
ARMC1NanoCT.DSU.PPU_core0.busslave	PVBusSlave
ARMC1NanoCT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMC1NanoCT.DSU.mpam_busslave	PVBusSlave
ARMC1NanoCT.DSU.shared_cache	PVCache
ARMC1NanoCT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMC1NanoCT.DSU.shared_cache.upstream[Y] (where Y = 0-4)	PVBusSlave
ARMC1NanoCT.DSU.utility_slave[0]	PVBusSlave
ARMC1NanoCT.MMAP	PVBusLogger
ARMC1NanoCT.MMAP.mapper	PVBusMapper
ARMC1NanoCT.RAS	PVBusLogger
ARMC1NanoCT.RAS.mapper	PVBusMapper
ARMC1NanoCT.cpu0	ARM_C1-Nano
ARMC1NanoCT.cpu0.UTLB	TLB
ARMC1NanoCT.cpu0.debug_rom	debug_rom
ARMC1NanoCT.cpu0.dtlb	TLB
ARMC1NanoCT.cpu0.l1dcache	PVCache
ARMC1NanoCT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMC1NanoCT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMC1NanoCT.cpu0.l1licache	PVCache
ARMC1NanoCT.cpu0.l1licache.downstream[0].pvbusmaster	PVBusMaster
ARMC1NanoCT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMC1NanoCT.cpu0.l2cache	PVCache
ARMC1NanoCT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMC1NanoCT.cpu0.l2cache.upstream[U] (where U = 0-1)	PVBusSlave
ARMC1NanoCT.ext_bus	PVBusLogger
ARMC1NanoCT.ext_bus.mapper	PVBusMapper
ARMC1NanoCT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

Name	Instance type
ARMC1NanoCT.global_debug_rom	debug_rom
ARMC1NanoCT.secondary_debug_rom	debug_rom
ARMC1NanoCT.sve	ScalableVectorExtension

This model has the following MTI trace components:

Name	Component type
ARMC1NanoCT.AMU	PVBusLogger
ARMC1NanoCT.AMU.mapper	PVBusMapper
ARMC1NanoCT.DAP	PVBusLogger
ARMC1NanoCT.DAP.mapper	PVBusMapper
ARMC1NanoCT.DSU	C1-DSU
ARMC1NanoCT.DSU.PPU_cluster	PPUv1
ARMC1NanoCT.DSU.PPU_cluster.busslave	PVBusSlave
ARMC1NanoCT.DSU.PPU_core0	PPUv1
ARMC1NanoCT.DSU.PPU_core0.busslave	PVBusSlave
ARMC1NanoCT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMC1NanoCT.DSU.mpam_busslave	PVBusSlave
ARMC1NanoCT.DSU.shared_cache	PVCache
ARMC1NanoCT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMC1NanoCT.DSU.shared_cache.upstream[Y] (where Y = 0-4)	PVBusSlave
ARMC1NanoCT.DSU.utility_slave[0]	PVBusSlave
ARMC1NanoCT.MMAP	PVBusLogger
ARMC1NanoCT.MMAP.mapper	PVBusMapper
ARMC1NanoCT.RAS	PVBusLogger
ARMC1NanoCT.RAS.mapper	PVBusMapper
ARMC1NanoCT.cpu0	ARM_C1-Nano
ARMC1NanoCT.cpu0.UTLB	TLB
ARMC1NanoCT.cpu0.l1dcache	PVCache
ARMC1NanoCT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMC1NanoCT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMC1NanoCT.cpu0.l1icache	PVCache
ARMC1NanoCT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMC1NanoCT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMC1NanoCT.cpu0.l2cache	PVCache
ARMC1NanoCT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMC1NanoCT.cpu0.l2cache.upstream[U] (where U = 0-1)	PVBusSlave
ARMC1NanoCT.ext_bus	PVBusLogger
ARMC1NanoCT.ext_bus.mapper	PVBusMapper
ARMC1NanoCT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

## Ports for ARMC1NanoCT

Port	Direction	Protocol	Description
acp_s	slave	PVBus	AXI ACP subordinate port.
AEND0MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND2MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND3MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART2MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	slave	Signal	CHI defined pins.
broadcastcachemaint	slave	Signal	ACE defined pins.
broadcastouter	slave	Signal	ACE defined pins.
broadcastpersist	slave	Signal	CHI defined pins.
cfgend	slave	Signal	This signal is for EE bit initialisation.
cfgsdisable	slave	Signal	This signal disables write access to some secure Interrupt Controller registers.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcsn_pchannel	master	PChannel	Cluster PCSM signal
cluster_powerdown_out	master	Signal	DEPRECATED - An output from PPUs that informs thermal controller of the cluster power info
cluster_ppu_hw_stat	master	Value	Cluster PPU power state
clustercriticalirq	master	Signal	Cluster Critical Irq
clustererrirq	master	Signal	Cluster Error Irq
clusterfaultirq	master	Signal	Cluster Fault Irq
clusterid	slave	Value	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set ManagerID64 of the core.

Port	Direction	Protocol	Description
clusterpmuirq	master	Signal	DynamiQ pmu irq
cme_pcs_m_pchannel	master	PChannel	Cluster PCSM signal
cmeerirq	master	Signal	RAS cme err irq
cme_faultirq	master	Signal	RAS cme fault irq
CNTHPIRQ	master	Signal	Timer signals to SOC.
CNTHPSIRQ	master	Signal	Timer signals to SOC
CNTHVIRQ	master	Signal	Timer signals to SOC.
CNTHVSIRQ	master	Signal	Timer signals to SOC
CNTPNSIRQ	master	Signal	Timer signals to SOC.
CNTPSIRQ	master	Signal	Timer signals to SOC.
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.
CNTVIRQ	master	Signal	Timer signals to SOC.
commirq	master	Signal	Interrupt signal from debug communications channel.
complexerrirq	master	Signal	Error indicator for an ECC error that causes potential data corruption or loss of coherency
complexfaultirq	master	Signal	Fault indicator for a detected 1 or 2 bit ECC error
core_clk_in	slave	ClockSignal	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
core_pcs_m_pchannel	master	PChannel	Core PCSM signals
core_powerdown_out	master	Signal	DEPRECATED - An output from PPUs that informs thermal controller of the core power info
core_ppu_hw_stat	master	Value	Notify the power state change inside the PPU.
coreerrirq	master	Signal	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq	master	Signal	Fault indicator for a detected 1 or 2 bit ECC error
coreinstrrun	master	Signal	Core Running State
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
ctidbgirq	master	Signal	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	slave	Signal	External debug interface.
dbgnopwrdown	master	Signal	No power-down request.
dbgprupreq	master	Signal	Debug power up request.
defaultMP	slave	Signal	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main manager interface ports.
dev_debug_s	slave	PVBus	External debug interface.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.
gicreset	master	Signal	An output from PPUs that can be used to reset GICCLK domain components

Port	Direction	Protocol	Description
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	slave	PVBus	External debug interface.
pmbirq	master	Signal	Interrupt signal from the statistical profiling unit.
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	master	Signal	PPU cluster interrupt.
ppu_cluster_isolate	master	Signal	PPU Cluster Domain isolation control.
ppu_cluster_wakerequest	slave	Signal	PPU cluster wake request signal.
ppu_cme_irq	master	Signal	PPU CME interrupt
ppu_core_irq	master	Signal	PPU core interrupt.
ppu_core_isolate	master	Signal	PPU Core Domain isolation control.
ppu_core_wakerequest	slave	Signal	PPU core wake request signal.
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	master	PVBus	The core will generate bus requests on this port.
pvbus_periph_m	master	PVBus	The core can generate peripheral bus request on this port.
rei	slave	Signal	Per core RAM Error Interrupt.
reset	slave	Signal	Reset signal to cluster.
rvbaraddr	slave	Value_64	Reset vector base address.
sci_m	master	SystemCoherencyInterface	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei	slave	Signal	Per core System Error physical pins.
spiden	slave	Signal	External debug interface.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq	master	Signal	Interrupt signal from the trace buffer unit.
utility_bus_s	slave	PVBus	Utility bus subordinate
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfiq	slave	Signal	Virtualised FIQ.
virq	slave	Signal	Virtualised IRQ.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Per core virtual System Error physical pins.

## Parameters for ARMC1NanoCT

### cpuX . CFGEND

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`cpuX.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`cpuX.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`cpuX.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**cpuX.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**cpuX.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**cpuX.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **cpuX.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

### **cpuX.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **cpuX.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

### **cpuX.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **cpuX.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

### **cpuX.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0



**cpuX.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**cpuX.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**cpuX.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**cpuX.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**cpuX.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**cpuX.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**cpuX.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**cpuX.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**cpuX.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**cpuX.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**cpuX.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**cpuX.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**cpuX.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**cpuX.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`cpuX.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`AEND0_DEFAULT`**

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

Type: `uint64_t`

Default value: `0x0`

### **`AEND1_DEFAULT`**

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

Type: `uint64_t`

Default value: `0x0`

### **`AEND2_DEFAULT`**

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

Type: `uint64_t`

Default value: `0x0`

### **`AEND3_DEFAULT`**

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

Type: `uint64_t`

Default value: `0x0`

### **`ASTART0_DEFAULT`**

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

Type: `uint64_t`

Default value: `0x0`

### **ASTART1\_DEFAULT**

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

Type: `uint64_t`

Default value: `0x0`

### **ASTART2\_DEFAULT**

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

Type: `uint64_t`

Default value: `0x0`

### **ASTART3\_DEFAULT**

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

Type: `uint64_t`

Default value: `0x0`

### **BROADCASTATOMIC**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`

Default value: `true`

### **BROADCASTCACHEMAINT**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

Type: `bool`

Default value: `false`

### **BROADCASTOUTER**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

Type: `bool`

Default value: `false`

### **BROADCASTPERSIST**

Enable broadcasting of cache clean to the point of persistence operations. The `broadcastpersist` signal will override this value if used.

Type: `bool`

Default value: `true`

### **CLUSTER\_ID**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the ManagerID64 of the core.

Type: `uint32_t`

Default value: `0x0`

### **CMO\_broadcast\_when\_cache\_state\_modelling\_disabled**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. This value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required.

Type: `uint8_t`

Default value: `1`

### **CPUCFR**

Value of CPU Configuration Register.

Type: `uint64_t`

Default value: `0x0`

### **DBGROMADDR**

Initialization value of DBGDRAR register. Bits[55:12] of this register specify the ROM table physical address.

Type: `uint64_t`

Default value: 0x0

**DBGROMADDRV**

If true, set bits[1:0] of the CP15 DBGDRAR to indicate that the address is valid.

Type: bool

Default value: false

**DSU.cme.CMECFR**

Value of CMECFR\_EL1 fields (ECC, CHI) for all CMEs.

Type: string

Default value: "{ \"ECC\": 0, \"CHI\": 0 }"

**DSU.cme.mpmm\_accumulator\_multiplier**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator. value of n means the accumulator will use (n \* accumulator value) to calculate the mpmm threshold (MPMM). is provided as a fast model workaround to handle cases where execution of quantum in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

Type: uint8\_t

Default value: 1

**DSU.cme.power\_on\_by\_default**

If true, CME PPU's will initialize in Dynamic mode out of reset, enabling the CME to power itself on/off automatically as it is used.

Type: bool

Default value: true

**GICDISABLE**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: bool

Default value: true

**NUM\_CMES**

Defines how many CMEs are associated with each cluster. This parameter must be in sync with SVE.ScalableVectorExtension SME configuration.

Type: uint8\_t

Default value: 0

### **NUM\_CORES**

Number of cores per cluster.

Type: uint8\_t

Default value: 1

### **bus\_type**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

Type: uint8\_t

Default value: 0

### **core\_cache\_protection**

core\_cache\_protection can change ERROFR, ERROPFGF and ERROPFGCTL fields. Possible values are: -1:Not implemented (by default), 0:Disabled, 1:Enabled.

Type: int8\_t

Default value: 1

### **core\_complex\_mapping**

Defines Complex descriptions for platforms that support several Cores per Complex like Cortex-A510. JSON format: {"complex0": { "cores" : [ 0, 1 ], "l2-cache" : {"exists":1, "size":16MB}}, ... , "complexN": { "cores" : [<core\_list>], "l2-cache" : {"exists":1, "size":16MB}}} where <core\_list> is the list of cores in the complexN. Effective only when the parameter value is not empty.

Type: string

Default value: "{ \"complex0\": { \"cores\": [0, 1], \"l2-cache\" : { \"exists\":1, \"size \": \"16MB\" } }, \"complex1\": { \"cores\": [2, 3], \"l2-cache\" : { \"exists\":1, \"size \": \"16MB\" } }, \"complex2\": { \"cores\": [4, 5], \"l2-cache\" : { \"exists\":1, \"size \": \"16MB\" } }, \"complex3\": { \"cores\": [6, 7], \"l2-cache\" : { \"exists\":1, \"size \": \"16MB\" } }, \"complex4\": { \"cores\": [8, 9], \"l2-cache\" : { \"exists\":1, \"size \": \"16MB\" } }, \"complex5\": { \"cores\": [10, 11], \"l2-cache\" : { \"exists\":1, \"size \": \"16MB\" } }, \"complex6\": { \"cores\": [12, 13], \"l2-cache\" : { \"exists\":1, \"size \": \"16MB\" } } }

### **core\_power\_on\_by\_default**

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPUs which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

Type: bool

Default value: `false`

**`cpi_div`**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**`cpi_mul`**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**`dcache-hit_latency`**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**`dcache-maintenance_latency`**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**`dcache-miss_latency`**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**`dcache-prefetch_enabled`**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

Type: `bool`

Default value: `false`



**dcache-read\_access\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**dcache-size**

L1 D-Cache size in bytes.

Type: uint32\_t

Default value: 0x8000

**dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**dcache-state\_modelled**

Set whether D-cache has stateful implementation.

Type: bool

Default value: false

**dcache-write\_access\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be

used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`dcache-write_latency`**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`enable_simulation_performance_optimizations`**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of `stage12_tlb_size` parameter to 1024).

Type: `bool`

Default value: `true`

### **`ete.CLAIMTAGS`**

Number of claim tags.

Type: `uint8_t`

Default value: 4

### **`ete.MAX_INST_PER_Q`**

Maximum limit for the number of instructions implied by a Q element.

Type: `uint16_t`

Default value: `0x1`

### **`ete.PIDR_CM0D`**

TRCPIDR CM0D value.

Type: `uint8_t`

Default value: 0

**ete.PIDR\_REVAND**

TRCPIDR REVAND value.

Type: `uint8_t`

Default value: 0

**ete.PIDR\_REVISION**

TRCPIDR REVISION value.

Type: `uint8_t`

Default value: 0

**ete.Q\_CADENCE**

Number of instruction blocks traced between two Q elements.

Type: `uint16_t`

Default value: 0x1

**ete.RES0\_STATEFUL**

Whether **RES0** bits are stateful or **RAZ/WI**.

Type: `bool`

Default value: `false`

**ete.RETSTACK**

Return stack depth.

Type: `uint8_t`

Default value: 3

**ete.REVISION**

TRCIDR1 revision value.

Type: `uint8_t`

Default value: 0

**ete.SIM\_OVERFLOW\_GRANULARITY**

Number of instruction blocks in each granule, for simulated overflow.

Type: `uint32_t`

Default value: 0x64

**ete.SIM\_OVERFLOW\_PERCENTAGE**

Percentage of instruction blocks lost in each granule, for simulated overflow.

Type: `uint8_t`

Default value: 0

**ete.SOURCE\_ADDRESS**

Allow generation of source address elements.

Type: `bool`

Default value: `false`

**ete.TRACE\_OUTPUT**

File to which to write trace byte stream.

Type: `string`

Default value: N/A

**ete.TRCSRSTA\_FORCED\_EXCEP**

TRCSRSTA value for a forcibly traced exception.

Type: `bool`

Default value: `false`

**force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks**

Force MTE tag accesses to **RAZ/WI** and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

Type: `bool`

Default value: `false`

**force\_zero\_PSTATE\_PAN**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

**force\_zero\_mpam\_partid\_and\_pmg**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

Type: `uint8_t`

Default value: `0`

**has\_actlr2**

If true ACLTR2 exists and ACTLR2(NS) is aliased to ACTLR\_EL1[63:32].

Type: `bool`

Default value: `true`

**has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

**has\_dot\_product**

Possible values of this parameter are:- 1, feature not implemented.- 2, feature is implemented.

Type: `uint8_t`

Default value: `2`

**has\_ete**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (`-plugin` or `-P`).

Type: `bool`

Default value: `false`

**has\_peripheral\_port**

If true, an additional AXI peripheral port is configured.

Type: `bool`

Default value: `false`

### **`icache-hit_latency`**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`icache-maintenance_latency`**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`icache-miss_latency`**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`icache-prefetch_enabled`**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

### **`icache-read_access_latency`**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-size**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: `0x8000`

**icache-state\_modelled**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

**invalidate\_code\_cache\_on\_icache\_cmo**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: `uint8_t`

Default value: `0`

**l3cache-has\_mpam**

L3 Cache has MPAM support.

Type: `bool`

Default value: `true`

**l3cache-hit\_latency**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-maintenance\_latency**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-miss\_latency**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-mpamf.max\_partid\_ns**

Maximum value of non-secure PARTID supported for DSU L3Cache. Options are 7 or 63.

Type: `uint8_t`

Default value: `63`

**l3cache-read\_access\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size `l3cache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-read\_bus\_width\_in\_bytes**

L3 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: `uint32_t`

Default value: `0x10`

**l3cache-read\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l3cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`



Default value: 0x0

### **l3cache-size**

L3 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

### **l3cache-snoop\_data\_transfer\_latency**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **l3cache-snoop\_issue\_latency**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **l3cache-ways**

L3 Cache number of ways (sets are implicit from size).

Type: uint16\_t

Default value: 16

### **l3cache-write\_access\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size l3cache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **l3cache-write\_bus\_width\_in\_bytes**

L3 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**l3cache-write\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. l3cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**memory\_tagging\_support\_level**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

Type: uint8\_t

Default value: 3

**mpmm\_accumulator\_multiplier**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator.value of n means the accumulator will use (n \* accumulator value) to calculate the mpmm threshold (MPMM).is provided as a fast model workaround to handle cases where execution of quantum in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

Type: uint8\_t

Default value: 1

**num\_acp**

Number of ACP ports.

Type: uint8\_t

Default value: 0

**num\_nodes**

Number of transport nodes. Zero implies direct-connect configuration.

Type: uint8\_t

Default value: 1

**pmu-num\_counters**

Number of PMU counters implemented.

Type: uint8\_t

Default value: 6

**ptw\_latency**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

**ras\_pfg\_clock\_mhz**

RAS Pseudo-Fault generation clock rate in MHz.

Type: `uint8_t`

Default value: 12

**scu\_cache\_protection**

SCU-L3 is configured with ECC if true.

Type: `bool`

Default value: `true`

**tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

**tlbi\_stall\_enabled**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: `bool`

Default value: `false`

**treat\_PAC\_as\_NOP**

Non-architecture parameter. Treat Pointer Authentication as **NOP**. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are **NOP**. Branch, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

Type: `bool`

Default value: `false`

**walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

## 3.33 ARMC1NanoCT\_C1ProCT

Defined in `LISA/ARMC1NanoCT_C1ProCT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
C1Nano r0p0	Preliminary support
C1Pro r0p0	Preliminary support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following parameters were added:

- `num_acp`
- `subcluster0.has_delayed_dbgreg`
- `subcluster0.has_delayed_sysreg`
- `subcluster1.has_delayed_dbgreg`
- `subcluster1.has_delayed_sysreg`

### About ARMC1NanoCT\_C1ProCT

The number of cores in each subcluster is configurable using the following parameters:

**subcluster0.NUM\_CORES**

Possible values are 1-13 (ARMC1NanoCT).

**subcluster1.NUM\_CORES**

Possible values are 1-13 (ARMC1ProCT).

The total number of cores in the cluster cannot exceed 14.

Port arrays are expanded to the width needed for the maximum number of cores in each subcluster. Use the following port array indexes:

- `<port_name>[0-12]` for cores in `subcluster0`.
- `<port_name>[13-25]` for cores in `subcluster1`.



All instances in the Master cross trigger matrix port array, `cti[26]` must be connected, regardless of the `NUM_CORES` value used.

Core-specific parameters have the following prefixes:

- `subcluster0.cpu0` to `subcluster0.cpu12` identify cores in `subcluster0`.
- `subcluster1.cpu0` to `subcluster1.cpu12` identify cores in `subcluster1`.

For information about the cores in this model, see:

- [ARMC1NanoCT](#).
- [ARMC1ProCT](#).

### Iris and MTI instances for ARMC1NanoCT\_C1ProCT

This model has the following Iris instances:

Name	Instance type
ARMC1NanoCT_C1ProCT	Cluster_ARM_C1-Nano_C1-PRO_Heterogeneous
ARMC1NanoCT_C1ProCT.AMU	PVBusLogger
ARMC1NanoCT_C1ProCT.AMU.mapper	PVBusMapper
ARMC1NanoCT_C1ProCT.DAP	PVBusLogger
ARMC1NanoCT_C1ProCT.DAP.mapper	PVBusMapper
ARMC1NanoCT_C1ProCT.DSU	C1-DSU
ARMC1NanoCT_C1ProCT.DSU.PPU_cluster	PPUv1
ARMC1NanoCT_C1ProCT.DSU.PPU_cluster.busslave	PVBusSlave
ARMC1NanoCT_C1ProCT.DSU.PPU_coreZ (where Z = 0-1)	PPUv1
ARMC1NanoCT_C1ProCT.DSU.PPU_coreZ.busslave (where Z = 0-1)	PVBusSlave
ARMC1NanoCT_C1ProCT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMC1NanoCT_C1ProCT.DSU.mpam_busslave	PVBusSlave
ARMC1NanoCT_C1ProCT.DSU.shared_cache	PVCache
ARMC1NanoCT_C1ProCT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMC1NanoCT_C1ProCT.DSU.shared_cache.upstream[Z] (where Z = 0-6)	PVBusSlave
ARMC1NanoCT_C1ProCT.DSU.utility_slave[0]	PVBusSlave
ARMC1NanoCT_C1ProCT.MMAP	PVBusLogger
ARMC1NanoCT_C1ProCT.MMAP.mapper	PVBusMapper
ARMC1NanoCT_C1ProCT.RAS	PVBusLogger
ARMC1NanoCT_C1ProCT.RAS.mapper	PVBusMapper
ARMC1NanoCT_C1ProCT.cpuZ.debug_rom (where Z = 0-1)	debug_rom
ARMC1NanoCT_C1ProCT.ext_bus	PVBusLogger
ARMC1NanoCT_C1ProCT.ext_bus.mapper	PVBusMapper
ARMC1NanoCT_C1ProCT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

Name	Instance type
ARMC1NanoCT_C1ProCT.global_debug_rom	debug_rom
ARMC1NanoCT_C1ProCT.secondary_debug_rom	debug_rom
ARMC1NanoCT_C1ProCT.subcluster0	Subcluster_ARM_C1-Nano
ARMC1NanoCT_C1ProCT.subcluster0.cpu0	ARM_C1-Nano
ARMC1NanoCT_C1ProCT.subclusterZ.cpu0.UTLB (where Z = 0-1)	TLB
ARMC1NanoCT_C1ProCT.subclusterZ.cpuU.dtlb (where Z = 0-1; U = 0-1)	TLB
ARMC1NanoCT_C1ProCT.subclusterZ.cpu0.l1dcache (where Z = 0-1)	PVCache
ARMC1NanoCT_C1ProCT.subclusterZ.cpu0.l1dcache.downstream[0].pvbusmaster (where Z = 0-1)	PVBusMaster
ARMC1NanoCT_C1ProCT.subclusterZ.cpu0.l1dcache.upstream[0] (where Z = 0-1)	PVBusSlave
ARMC1NanoCT_C1ProCT.subclusterZ.cpu0.l1icache (where Z = 0-1)	PVCache
ARMC1NanoCT_C1ProCT.subclusterZ.cpu0.l1icache.downstream[0].pvbusmaster (where Z = 0-1)	PVBusMaster
ARMC1NanoCT_C1ProCT.subclusterZ.cpu0.l1icache.upstream[0] (where Z = 0-1)	PVBusSlave
ARMC1NanoCT_C1ProCT.subclusterZ.cpu0.l2cache (where Z = 0-1)	PVCache
ARMC1NanoCT_C1ProCT.subclusterZ.cpu0.l2cache.downstream[0].pvbusmaster (where Z = 0-1)	PVBusMaster
ARMC1NanoCT_C1ProCT.subclusterZ.cpu0.l2cache.upstream[W] (where Z = 0-1; W = 0-1)	PVBusSlave
ARMC1NanoCT_C1ProCT.subclusterZ.sve (where Z = 0-1)	ScalableVectorExtension
ARMC1NanoCT_C1ProCT.subcluster1	Subcluster_ARM_C1-Pro
ARMC1NanoCT_C1ProCT.subcluster1.cpu0	ARM_C1-Pro

This model has the following MTI trace components:

Name	Component type
ARMC1NanoCT_C1ProCT.AMU	PVBusLogger
ARMC1NanoCT_C1ProCT.AMU.mapper	PVBusMapper
ARMC1NanoCT_C1ProCT.DAP	PVBusLogger
ARMC1NanoCT_C1ProCT.DAP.mapper	PVBusMapper
ARMC1NanoCT_C1ProCT.DSU	C1-DSU
ARMC1NanoCT_C1ProCT.DSU.PPU_cluster	PPUv1
ARMC1NanoCT_C1ProCT.DSU.PPU_cluster.busslave	PVBusSlave
ARMC1NanoCT_C1ProCT.DSU.PPU_coreZ (where Z = 0-1)	PPUv1
ARMC1NanoCT_C1ProCT.DSU.PPU_coreZ.busslave (where Z = 0-1)	PVBusSlave
ARMC1NanoCT_C1ProCT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMC1NanoCT_C1ProCT.DSU.mpam_busslave	PVBusSlave
ARMC1NanoCT_C1ProCT.DSU.shared_cache	PVCache
ARMC1NanoCT_C1ProCT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMC1NanoCT_C1ProCT.DSU.shared_cache.upstream[Z] (where Z = 0-6)	PVBusSlave
ARMC1NanoCT_C1ProCT.DSU.utility_slave[0]	PVBusSlave

Name	Component type
ARMC1NanoCT_C1ProCT.MMAP	PVBusLogger
ARMC1NanoCT_C1ProCT.MMAP.mapper	PVBusMapper
ARMC1NanoCT_C1ProCT.RAS	PVBusLogger
ARMC1NanoCT_C1ProCT.RAS.mapper	PVBusMapper
ARMC1NanoCT_C1ProCT.ext_bus	PVBusLogger
ARMC1NanoCT_C1ProCT.ext_bus.mapper	PVBusMapper
ARMC1NanoCT_C1ProCT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMC1NanoCT_C1ProCT.subcluster0.cpu0	ARM_C1-Nano
ARMC1NanoCT_C1ProCT.subclusterZ.cpu0.UTLB (where Z = 0-1)	TLB
ARMC1NanoCT_C1ProCT.subclusterZ.cpu0.l1dcache (where Z = 0-1)	PVCache
ARMC1NanoCT_C1ProCT.subclusterZ.cpu0.l1dcache.downstream[0].pvbusmaster (where Z = 0-1)	PVBusMaster
ARMC1NanoCT_C1ProCT.subclusterZ.cpu0.l1dcache.upstream[0] (where Z = 0-1)	PVBusSlave
ARMC1NanoCT_C1ProCT.subclusterZ.cpu0.l1icache (where Z = 0-1)	PVCache
ARMC1NanoCT_C1ProCT.subclusterZ.cpu0.l1icache.downstream[0].pvbusmaster (where Z = 0-1)	PVBusMaster
ARMC1NanoCT_C1ProCT.subclusterZ.cpu0.l1icache.upstream[0] (where Z = 0-1)	PVBusSlave
ARMC1NanoCT_C1ProCT.subclusterZ.cpu0.l2cache (where Z = 0-1)	PVCache
ARMC1NanoCT_C1ProCT.subclusterZ.cpu0.l2cache.downstream[0].pvbusmaster (where Z = 0-1)	PVBusMaster
ARMC1NanoCT_C1ProCT.subclusterZ.cpu0.l2cache.upstream[W] (where Z = 0-1; W = 0-1)	PVBusSlave
ARMC1NanoCT_C1ProCT.subcluster1.cpu0	ARM_C1-Pro

### Ports for ARMC1NanoCT\_C1ProCT

Port	Direction	Protocol	Description
acp_s	slave	PVBus	AXI ACP subordinate port.
AEND0MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND2MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND3MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART2MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	slave	Signal	CHI defined pins.

Port	Direction	Protocol	Description
broadcastcachemaint	slave	Signal	ACE defined pins.
broadcastrouter	slave	Signal	ACE defined pins.
broadcastpersist	slave	Signal	CHI defined pins.
cfgend	slave	Signal	This signal is for EE bit initialisation.
cfgsdisable	slave	Signal	This signal disables write access to some secure Interrupt Controller registers.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcs_m_pchannel	master	PChannel	Cluster PCSM signal
cluster_powerdown_out	master	Signal	DEPRECATED - An output from PPU that informs thermal controller of the cluster power info
cluster_ppu_hw_stat	master	Value	Cluster PPU power state
clusterid	slave	Value	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set ManagerID64 of the core.
clusterpmuirq	master	Signal	DynamiQ pmu irq
cme_pcs_m_pchannel	master	PChannel	Cluster PCSM signal
cmeerrirq	master	Signal	RAS cme err irq
cmefaultirq	master	Signal	RAS cme fault irq
CNTHPIRQ	master	Signal	Timer signals to SOC.
CNTHPSIRQ	master	Signal	Timer signals to SOC
CNTHVIRQ	master	Signal	Timer signals to SOC.
CNTHVSIRQ	master	Signal	Timer signals to SOC
CNTPNSIRQ	master	Signal	Timer signals to SOC.
CNTPSIRQ	master	Signal	Timer signals to SOC.
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.
CNTVIRQ	master	Signal	Timer signals to SOC.
commirq	master	Signal	Interrupt signal from debug communications channel.
core_clk_in	slave	ClockSignal	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
core_pcs_m_pchannel	master	PChannel	Core PCSM signals
core_powerdown_out	master	Signal	DEPRECATED - An output from PPU that informs thermal controller of the core power info
core_ppu_hw_stat	master	Value	Notify the power state change inside the PPU.



Port	Direction	Protocol	Description
coreerrirq	master	Signal	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq	master	Signal	Fault indicator for a detected 1 or 2 bit ECC error
coreinstrrun	master	Signal	Core Running State
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
ctidbgirq	master	Signal	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	slave	Signal	External debug interface.
dbgnopwrdown	master	Signal	No power-down request.
dbgpwrupreq	master	Signal	Debug power up request.
defaultMP	slave	Signal	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main manager interface ports.
dev_debug_s	slave	PVBus	External debug interface.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.
gicreset	master	Signal	An output from PPU that can be used to reset GICCLK domain components
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	slave	PVBus	External debug interface.
pmbirq	master	Signal	Interrupt signal from the statistical profiling unit.
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	master	Signal	PPU cluster interrupt.
ppu_cluster_isolate	master	Signal	PPU Cluster Domain isolation control.
ppu_cluster_wakerequest	slave	Signal	PPU cluster wake request signal.
ppu_cme_irq	master	Signal	PPU CME interrupt
ppu_core_irq	master	Signal	PPU core interrupt.
ppu_core_isolate	master	Signal	PPU Core Domain isolation control.
ppu_core_wakerequest	slave	Signal	PPU core wake request signal.
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	master	PVBus	The core will generate bus requests on this port.
pvbus_periph_m	master	PVBus	The core can generate peripheral bus request on this port.
rei	slave	Signal	Per core RAM Error Interrupt.
reset	slave	Signal	Reset signal to cluster.
rvbaraddr	slave	Value_64	Reset vector base address.
sci_m	master	SystemCoherencyInterface	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain

Port	Direction	Protocol	Description
sei	slave	Signal	Per core System Error physical pins.
spiden	slave	Signal	External debug interface.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq	master	Signal	Interrupt signal from the trace buffer unit.
utility_bus_s	slave	PVBus	Utility bus subordinate
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfiq	slave	Signal	Virtualised FIQ.
virq	slave	Signal	Virtualised IRQ.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Per core virtual System Error physical pins.

## Parameters for ARMC1NanoCT\_C1ProCT

### AEND0\_DEFAULT

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

Type: uint64\_t

Default value: 0x0

### AEND1\_DEFAULT

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

Type: uint64\_t

Default value: 0x0

### AEND2\_DEFAULT

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

Type: uint64\_t

Default value: 0x0

### AEND3\_DEFAULT

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

Type: uint64\_t

Default value: 0x0

#### **ASTART0\_DEFAULT**

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

Type: `uint64_t`

Default value: 0x0

#### **ASTART1\_DEFAULT**

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

Type: `uint64_t`

Default value: 0x0

#### **ASTART2\_DEFAULT**

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

Type: `uint64_t`

Default value: 0x0

#### **ASTART3\_DEFAULT**

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

Type: `uint64_t`

Default value: 0x0

#### **BROADCASTATOMIC**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`

Default value: `true`

#### **BROADCASTCACHEMAINT**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

Type: `bool`

Default value: `false`

### **BROADCASTOUTER**

Enable broadcasting of Outer Shareable transactions. The `broadcastouter` signal will override this value if used.

Type: `bool`

Default value: `false`

### **BROADCASTPERSIST**

Enable broadcasting of cache clean to the point of persistence operations. The `broadcastpersist` signal will override this value if used.

Type: `bool`

Default value: `true`

### **CLUSTER\_ID**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the ManagerID64 of the core.

Type: `uint32_t`

Default value: `0x0`

### **CMO\_broadcast\_when\_cache\_state\_modelling\_disabled**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. This value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required.

Type: `uint8_t`

Default value: `1`

### **DBGROMADDR**

Initialization value of DBGDRAR register. Bits[55:12] of this register specify the ROM table physical address.

Type: `uint64_t`

Default value: 0x0

### **DBGROMADDRV**

If true, set bits[1:0] of the CP15 DBGDRAR to indicate that the address is valid.

Type: bool

Default value: false

### **GICDISABLE**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: bool

Default value: true

### **NUM\_CMES**

Defines how many CMEs are associated with each cluster. This parameter must be in sync with SVE.ScalableVectorExtension SME configuration.

Type: uint8\_t

Default value: 0

### **bus\_type**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

Type: uint8\_t

Default value: 0

### **core\_complex\_mapping**

Defines Complex descriptions for platforms that support several Cores per Complex like Cortex-A510. JSON format: {"complex0": { "cores" : [ 0, 1 ], "l2-cache" : {"exists":1, "size":16MB}}, ... , "complexN": { "cores" : [<core\_list>, "l2-cache" : {"exists":1, "size":16MB}}} where <core\_list> is the list of cores in the complexN. Effective only when the parameter value is not empty.

Type: string

Default value: "{ \"complex0\": { \"cores\": [0, 1], \"l2-cache\" : {\"exists\":1, \"size\": \"16MB\"}}, \"complex1\": { \"cores\": [2, 3], \"l2-cache\" : {\"exists\":1, \"size\": \"16MB\"}}, \"complex2\": { \"cores\": [4, 5], \"l2-cache\" : {\"exists\":1, \"size\": \"16MB\"}}, \"complex3\": { \"cores\": [6, 7], \"l2-cache\" : {\"exists\":1, \"size\": \"16MB\"}}, \"complex4\": { \"cores\": [8, 9], \"l2-cache\" : {\"exists\":1, \"size\": \"16MB\"}}, \"complex5\": { \"cores\": [10, 11], \"l2-cache\" : {\"exists\":1, \"size\": \"16MB\"}}, \"complex6\": { \"cores\": [12, 13], \"l2-cache\" : {\"exists\":1, \"size\": \"16MB\"}} }"

**core\_power\_on\_by\_default**

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPUs which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

Type: `bool`

Default value: `false`

**dcache-state\_modelled**

Set whether D-cache has stateful implementation.

Type: `bool`

Default value: `false`

**enable\_simulation\_performance\_optimizations**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of `stage12_tlb_size` parameter to 1024).

Type: `bool`

Default value: `true`

**has\_peripheral\_port**

If true, an additional AXI peripheral port is configured.

Type: `bool`

Default value: `false`

**icache-state\_modelled**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

**l3cache-has\_mpam**

L3 Cache has MPAM support.

Type: `bool`

Default value: `true`

**l3cache-hit\_latency**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-maintenance\_latency**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-miss\_latency**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-mpamf.max\_partid\_ns**

Maximum value of non-secure PARTID supported for DSU L3Cache. Options are 7 or 63.

Type: `uint8_t`

Default value: 63

**l3cache-read\_access\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size `l3cache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-read\_bus\_width\_in\_bytes**

L3 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: `uint32_t`

Default value: `0x10`

**l3cache-read\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l3cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**l3cache-size**

L3 Cache size in bytes.

Type: `uint32_t`

Default value: 0x80000

**l3cache-snoop\_data\_transfer\_latency**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**l3cache-snoop\_issue\_latency**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**l3cache-ways**

L3 Cache number of ways (sets are implicit from size).

Type: `uint16_t`

Default value: 16

**l3cache-write\_access\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size `l3cache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`



Default value: 0x0

### **l3cache-write\_bus\_width\_in\_bytes**

L3 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

### **l3cache-write\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. l3cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **mpmm\_accumulator\_multiplier**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator. value of n means the accumulator will use (n \* accumulator value) to calculate the mpmm threshold (MPMM). is provided as a fast model workaround to handle cases where execution of quantum in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

Type: uint8\_t

Default value: 1

### **num\_acp**

Number of ACP ports.

Type: uint8\_t

Default value: 0

### **num\_nodes**

Number of transport nodes. Zero implies direct-connect configuration.

Type: uint8\_t

Default value: 1

### **scu\_cache\_protection**

SCU-L3 is configured with ECC if true.

Type: bool

Default value: `true`

### **subcluster0.CPUCFR**

Value of CPU Configuration Register.

Type: `uint64_t`

Default value: `0x0`

### **subcluster0.NUM\_CORES**

Number of cores in the subcluster. Total number of cores in cluster may not exceed 14.

Type: `uint8_t`

Default value: `1`

### **subcluster0.core\_cache\_protection**

`core_cache_protection` can change `ERROFR`, `ERROPFGF` and `ERROPFGCTL` fields. Possible values are: -1:Not implemented (by default), 0:Disabled, 1:Enabled.

Type: `int8_t`

Default value: `1`

### **subcluster0.cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

### **subcluster0.cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

### **subcluster0.cpu0.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster0.cpu0.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster0.cpu0.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu0.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu0.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu0.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu0.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu0.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu0.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu0.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster0.cpu0.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

### **subcluster0.cpu0.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster0.cpu0.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

### **subcluster0.cpu0.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster0.cpu0.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

### **subcluster0.cpu0.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu0.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu0.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu0.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu0.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster0.cpu0.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster0.cpu0.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu0.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu0.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu0.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu0.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu0.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu0.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu0.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu0.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu0.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu1.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu1.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu1.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu1.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu1.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`



**subcluster0.cpu1.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu1.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu1.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu1.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: `uint32_t`

Default value: `0x10`

**subcluster0.cpu1.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu1.l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: 0x80000

**subcluster0.cpu1.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu1.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu1.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu1.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu1.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu1.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu1.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu1.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster0.cpu1.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

### **`subcluster0.cpu1.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **`subcluster0.cpu1.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **`subcluster0.cpu1.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster0.cpu1.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu1.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu1.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster0.cpu1.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

#### **`subcluster0.cpu1.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster0.cpu1.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu1.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster0.cpu10.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu10.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu10.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: 0x0

**subcluster0.cpu10.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter trace\_special\_hlt\_imm16.

Type: bool

Default value: false

**subcluster0.cpu10.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu10.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu10.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu10.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu10.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

### **subcluster0.cpu10.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster0.cpu10.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

### **subcluster0.cpu10.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster0.cpu10.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster0.cpu10.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

### **subcluster0.cpu10.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu10.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu10.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu10.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu10.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu10.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu10.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000



**subcluster0.cpu10.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu10.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster0.cpu10.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster0.cpu10.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu10.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu10.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu10.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu10.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu10.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu10.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu10.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu10.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster0.cpu10.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster0.cpu11.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster0.cpu11.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster0.cpu11.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu11.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu11.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu11.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu11.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu11.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu11.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu11.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu11.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu11.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu11.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu11.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu11.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu11.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu11.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu11.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu11.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu11.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu11.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu11.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu11.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster0.cpu11.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster0.cpu11.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster0.cpu11.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster0.cpu11.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu11.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu11.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu11.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu11.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu11.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster0.cpu11.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu11.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster0.cpu12.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu12.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu12.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu12.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`



**subcluster0.cpu12.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu12.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu12.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu12.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu12.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu12.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu12.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu12.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu12.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu12.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu12.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu12.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu12.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu12.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu12.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu12.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu12.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu12.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu12.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**`subcluster0.cpu12.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**`subcluster0.cpu12.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**`subcluster0.cpu12.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**`subcluster0.cpu12.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**`subcluster0.cpu12.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**`subcluster0.cpu12.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**`subcluster0.cpu12.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster0.cpu12.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu12.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu12.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu12.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu2.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu2.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu2.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu2.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu2.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu2.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu2.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu2.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

#### **subcluster0.cpu2.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

#### **subcluster0.cpu2.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu2.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

#### **subcluster0.cpu2.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu2.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu2.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu2.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu2.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu2.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu2.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000



**subcluster0.cpu2.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu2.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu2.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster0.cpu2.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster0.cpu2.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu2.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu2.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu2.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu2.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster0.cpu2.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu2.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu2.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster0.cpu2.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu3.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu3.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu3.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu3.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu3.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu3.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu3.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu3.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu3.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu3.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu3.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu3.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu3.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu3.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu3.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu3.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster0.cpu3.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster0.cpu3.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster0.cpu3.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster0.cpu3.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu3.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu3.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu3.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu3.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu3.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster0.cpu3.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu3.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster0.cpu4.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu4.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu4.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu4.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`



**subcluster0.cpu4.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu4.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu4.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu4.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu4.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu4.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu4.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu4.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu4.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu4.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**`subcluster0.cpu4.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**`subcluster0.cpu4.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**`subcluster0.cpu4.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**`subcluster0.cpu4.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**`subcluster0.cpu4.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**`subcluster0.cpu4.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**`subcluster0.cpu4.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

#### **`subcluster0.cpu4.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

#### **`subcluster0.cpu4.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster0.cpu4.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu4.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster0.cpu5.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu5.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster0.cpu5.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu5.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu5.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu5.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu5.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu5.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster0.cpu5.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu5.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu5.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu5.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu5.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu5.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu5.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000



**subcluster0.cpu5.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu5.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu5.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster0.cpu5.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster0.cpu5.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu5.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu5.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu5.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu5.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster0.cpu5.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu5.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu5.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster0.cpu5.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **subcluster0.cpu6.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **subcluster0.cpu6.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **subcluster0.cpu6.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **subcluster0.cpu6.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **subcluster0.cpu6.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **subcluster0.cpu6.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu6.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu6.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu6.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu6.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu6.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu6.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu6.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu6.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu6.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu6.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster0.cpu6.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster0.cpu6.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster0.cpu6.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster0.cpu6.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu6.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu6.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu6.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu6.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu6.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster0.cpu6.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu6.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster0.cpu7.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu7.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu7.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu7.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`



**subcluster0.cpu7.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu7.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu7.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu7.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu7.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu7.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu7.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu7.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu7.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu7.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**`subcluster0.cpu7.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**`subcluster0.cpu7.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**`subcluster0.cpu7.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**`subcluster0.cpu7.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**`subcluster0.cpu7.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**`subcluster0.cpu7.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**`subcluster0.cpu7.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster0.cpu7.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu7.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu7.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu7.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu8.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu8.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster0.cpu8.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu8.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu8.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu8.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu8.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu8.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster0.cpu8.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu8.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu8.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu8.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu8.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu8.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu8.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000



**subcluster0.cpu8.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu8.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu8.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster0.cpu8.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster0.cpu8.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu8.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu8.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu8.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu8.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster0.cpu8.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu8.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu8.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster0.cpu8.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu9.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu9.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu9.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu9.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu9.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu9.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu9.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu9.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu9.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu9.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu9.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu9.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu9.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu9.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu9.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu9.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster0.cpu9.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster0.cpu9.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster0.cpu9.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster0.cpu9.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu9.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu9.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu9.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu9.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu9.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu9.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu9.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.dcache-hit_latency`**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.dcache-maintenance_latency`**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.dcache-miss_latency`**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.dcache-prefetch_enabled`**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

Type: `bool`



Default value: `false`

### **`subcluster0.dcache-read_access_latency`**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.dcache-read_latency`**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.dcache-size`**

L1 D-Cache size in bytes.

Type: `uint32_t`

Default value: `0x8000`

### **`subcluster0.dcache-snoop_data_transfer_latency`**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.dcache-write_access_latency`**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.ete.CLAIMTAGS**

Number of claim tags.

Type: uint8\_t

Default value: 4

**subcluster0.ete.MAX\_INST\_PER\_Q**

Maximum limit for the number of instructions implied by a Q element.

Type: uint16\_t

Default value: 0x1

**subcluster0.ete.PIDR\_CM0D**

TRCPIDR CM0D value.

Type: uint8\_t

Default value: 0

**subcluster0.ete.PIDR\_REVAND**

TRCPIDR REVAND value.

Type: uint8\_t

Default value: 0

**subcluster0.ete.PIDR\_REVISION**

TRCPIDR REVISION value.

Type: uint8\_t

Default value: 0

**subcluster0.ete.Q\_CADENCE**

Number of instruction blocks traced between two Q elements.

Type: `uint16_t`

Default value: `0x1`

#### **`subcluster0.ete.RES0_STATEFUL`**

Whether **RES0** bits are stateful or **RAZ/WI**.

Type: `bool`

Default value: `false`

#### **`subcluster0.ete.RETSTACK`**

Return stack depth.

Type: `uint8_t`

Default value: `3`

#### **`subcluster0.ete.REVISION`**

TRCIDR1 revision value.

Type: `uint8_t`

Default value: `0`

#### **`subcluster0.ete.SIM_OVERFLOW_GRANULARITY`**

Number of instruction blocks in each granule, for simulated overflow.

Type: `uint32_t`

Default value: `0x64`

#### **`subcluster0.ete.SIM_OVERFLOW_PERCENTAGE`**

Percentage of instruction blocks lost in each granule, for simulated overflow.

Type: `uint8_t`

Default value: `0`

#### **`subcluster0.ete.SOURCE_ADDRESS`**

Allow generation of source address elements.

Type: `bool`

Default value: `false`

#### **`subcluster0.ete.TRACE_OUTPUT`**

File to which to write trace byte stream.

Type: `string`

Default value: `N/A`

#### **`subcluster0.ete.TRCRSRTA_FORCED_EXCEP`**

TRCRSR.TA value for a forcibly traced exception.

Type: `bool`

Default value: `false`

#### **`subcluster0.force_mte_tag_access_razwi_and_ignore_tag_checks`**

Force MTE tag accesses to **RAZ/WI** and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

Type: `bool`

Default value: `false`

#### **`subcluster0.force_zero_PSTATE_PAN`**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

#### **`subcluster0.force_zero_mpam_partid_and_pmg`**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

Type: `uint8_t`

Default value: `0`

#### **`subcluster0.has_actlr2`**

If true ACLTR2 exists and ACTLR2(NS) is aliased to ACTLR\_EL1[63:32].

Type: `bool`

Default value: `true`

#### **`subcluster0.has_delayed_dbgreg`**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

### **`subcluster0.has_delayed_sysreg`**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

### **`subcluster0.has_dot_product`**

Possible values of this parameter are:- 1, feature not implemented.- 2, feature is implemented.

Type: `uint8_t`

Default value: 2

### **`subcluster0.has_ete`**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (-plugin or -P).

Type: `bool`

Default value: `false`

### **`subcluster0.icache-hit_latency`**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.icache-maintenance_latency`**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.icache-miss_latency`**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **subcluster0.icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

Type: bool

Default value: false

### **subcluster0.icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster0.icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster0.icache-size**

L1 I-Cache size in bytes.

Type: uint32\_t

Default value: 0x8000

### **subcluster0.invalidate\_code\_cache\_on\_icache\_cmo**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: uint8\_t

Default value: 0

**subcluster0.memory\_tagging\_support\_level**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

Type: `uint8_t`

Default value: 3

**subcluster0.pmu-num\_counters**

Number of PMU counters implemented.

Type: `uint8_t`

Default value: 6

**subcluster0.ptw\_latency**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

**subcluster0.ras\_pfg\_clock\_mhz**

RAS Pseudo-Fault generation clock rate in MHz.

Type: `uint8_t`

Default value: 12

**subcluster0.tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

**subcluster0.tlbi\_stall\_enabled**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: `bool`

Default value: `false`

**subcluster0.treat\_PAC\_as\_NOP**

Non-architecture parameter. Treat Pointer Authentication as **NOP**. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are **NOP**. Branch, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

Type: `bool`

Default value: `false`

**subcluster0.walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.CPUCFR**

Value of CPU Configuration Register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.NUM\_CORES**

Number of cores in the subcluster. Total number of cores in cluster may not exceed 14.

Type: `uint8_t`

Default value: `1`

**subcluster1.core\_cache\_protection**

`core_cache_protection` can change `ERROFR`, `ERROPFGF` and `ERROPFGCTL` fields. Possible values are: -1:Not implemented (by default), 0:Disabled, 1:Enabled.

Type: `int8_t`

Default value: `1`

**subcluster1.cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`



**subcluster1.cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**subcluster1.cpu0.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu0.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster1.cpu0.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu0.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu0.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu0.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

#### **`subcluster1.cpu0.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu0.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu0.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu0.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu0.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu0.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x100000`

#### **`subcluster1.cpu0.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu0.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu0.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu0.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu0.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

#### **`subcluster1.cpu0.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: 0

#### **`subcluster1.cpu0.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

#### **`subcluster1.cpu0.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

#### **`subcluster1.cpu0.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: 0x123456

#### **`subcluster1.cpu0.semihosting-T32_HLT`**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

#### **`subcluster1.cpu0.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

#### **`subcluster1.cpu0.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

#### **`subcluster1.cpu0.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

### **`subcluster1.cpu0.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster1.cpu0.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu0.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster1.cpu0.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster1.cpu0.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster1.cpu0.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu0.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu0.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster1.cpu1.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu1.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster1.cpu1.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu1.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu1.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu1.force-fpsid`**

Override the FPSID value.

Type: `bool`

Default value: `true`

### **`subcluster1.cpu1.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to

the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu1.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x100000`

#### **`subcluster1.cpu1.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu1.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu1.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu1.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`



**subcluster1.cpu1.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

**subcluster1.cpu1.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

**subcluster1.cpu1.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu1.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu1.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu1.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster1.cpu1.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster1.cpu1.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu1.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu1.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu1.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu1.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu1.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu1.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: 0xf000000

**subcluster1.cpu1.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu1.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster1.cpu1.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: true

**subcluster1.cpu10.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: false

**subcluster1.cpu10.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: bool

Default value: false

**subcluster1.cpu10.CRYPTODISABLE**

Disable cryptographic features.

Type: bool

Default value: false

**subcluster1.cpu10.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu10.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu10.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu10.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu10.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu10.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu10.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu10.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu10.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x100000

**subcluster1.cpu10.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu10.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu10.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu10.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu10.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

### **`subcluster1.cpu10.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

### **`subcluster1.cpu10.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu10.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu10.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu10.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster1.cpu10.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster1.cpu10.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster1.cpu10.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster1.cpu10.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu10.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu10.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu10.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu10.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu10.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu10.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu10.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster1.cpu11.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`



**subcluster1.cpu11.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster1.cpu11.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu11.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu11.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu11.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu11.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu11.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu11.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu11.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu11.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu11.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x100000`

### **`subcluster1.cpu11.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu11.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu11.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu11.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu11.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu11.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu11.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu11.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu11.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu11.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster1.cpu11.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster1.cpu11.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu11.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu11.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu11.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu11.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu11.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster1.cpu11.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu11.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu11.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster1.cpu11.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster1.cpu12.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu12.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu12.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu12.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu12.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu12.force-fpsid`**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu12.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu12.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu12.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu12.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu12.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu12.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x100000

#### **subcluster1.cpu12.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu12.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu12.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu12.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu12.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

#### **subcluster1.cpu12.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t



Default value: 0

**subcluster1.cpu12.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster1.cpu12.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster1.cpu12.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: 0x123456

**subcluster1.cpu12.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster1.cpu12.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster1.cpu12.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster1.cpu12.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster1.cpu12.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: true

**subcluster1.cpu12.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu12.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu12.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster1.cpu12.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu12.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu12.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu12.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster1.cpu2.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu2.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster1.cpu2.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu2.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu2.force-fpsid`**

Override the FPSID value.

Type: `bool`

Default value: `true`

### **`subcluster1.cpu2.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu2.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu2.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu2.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu2.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to

the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu2.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x100000`

#### **`subcluster1.cpu2.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu2.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu2.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu2.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu2.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu2.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu2.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu2.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu2.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster1.cpu2.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster1.cpu2.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu2.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu2.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu2.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu2.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu2.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: 0xf000000

**subcluster1.cpu2.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu2.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster1.cpu2.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: true

**subcluster1.cpu3.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: false

**subcluster1.cpu3.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: bool

Default value: false

**subcluster1.cpu3.CRYPTODISABLE**

Disable cryptographic features.

Type: bool

Default value: false



**subcluster1.cpu3.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu3.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu3.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x100000

**subcluster1.cpu3.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu3.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu3.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

### **`subcluster1.cpu3.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

### **`subcluster1.cpu3.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu3.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu3.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu3.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster1.cpu3.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster1.cpu3.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster1.cpu3.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster1.cpu3.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu3.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu3.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu3.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu3.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu3.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu3.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster1.cpu4.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu4.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster1.cpu4.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu4.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu4.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu4.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu4.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu4.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu4.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu4.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu4.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu4.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x100000`

#### **`subcluster1.cpu4.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu4.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu4.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu4.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu4.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu4.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu4.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000



**subcluster1.cpu4.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu4.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu4.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster1.cpu4.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster1.cpu4.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu4.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu4.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu4.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu4.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu4.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu4.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu4.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu4.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu4.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster1.cpu5.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu5.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu5.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu5.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu5.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu5.force-fpsid`**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu5.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu5.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu5.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu5.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu5.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu5.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x100000

#### **subcluster1.cpu5.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu5.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu5.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu5.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu5.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

#### **subcluster1.cpu5.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu5.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster1.cpu5.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster1.cpu5.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: 0x123456

**subcluster1.cpu5.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster1.cpu5.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster1.cpu5.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster1.cpu5.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster1.cpu5.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: true

**subcluster1.cpu5.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu5.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu5.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster1.cpu5.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu5.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu5.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu5.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster1.cpu6.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu6.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster1.cpu6.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu6.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu6.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.



Type: `bool`

Default value: `false`

### **`subcluster1.cpu6.force-fpsid`**

Override the FPSID value.

Type: `bool`

Default value: `true`

### **`subcluster1.cpu6.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu6.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu6.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu6.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu6.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to

the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu6.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x100000`

#### **`subcluster1.cpu6.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu6.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu6.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu6.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu6.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu6.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu6.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu6.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu6.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu6.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster1.cpu6.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster1.cpu6.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu6.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu6.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu6.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu6.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu6.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu6.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: 0xf000000

**subcluster1.cpu6.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu6.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster1.cpu6.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: true

**subcluster1.cpu7.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: false

**subcluster1.cpu7.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: bool

Default value: false

**subcluster1.cpu7.CRYPTODISABLE**

Disable cryptographic features.

Type: bool

Default value: false

**subcluster1.cpu7.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu7.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu7.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu7.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu7.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu7.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu7.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu7.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu7.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x100000

**subcluster1.cpu7.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu7.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu7.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu7.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu7.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

### **`subcluster1.cpu7.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

### **`subcluster1.cpu7.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu7.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu7.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`



**subcluster1.cpu7.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster1.cpu7.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster1.cpu7.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster1.cpu7.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster1.cpu7.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu7.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: 0x0

**subcluster1.cpu7.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: 0xf000000

**subcluster1.cpu7.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu7.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu7.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu7.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu7.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster1.cpu8.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu8.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster1.cpu8.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu8.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu8.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu8.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu8.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu8.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu8.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu8.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu8.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu8.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x100000`

### **`subcluster1.cpu8.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu8.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu8.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu8.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu8.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu8.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu8.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu8.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu8.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu8.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster1.cpu8.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster1.cpu8.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu8.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu8.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu8.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu8.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu8.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster1.cpu8.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu8.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu8.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster1.cpu8.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster1.cpu9.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster1.cpu9.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu9.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu9.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu9.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu9.force-fpsid`**

Override the FPSID value.

Type: `bool`

Default value: `true`



**subcluster1.cpu9.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x100000

**subcluster1.cpu9.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu9.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu9.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster1.cpu9.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster1.cpu9.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: 0x123456

**subcluster1.cpu9.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster1.cpu9.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster1.cpu9.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster1.cpu9.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster1.cpu9.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: true

**subcluster1.cpu9.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu9.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster1.cpu9.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu9.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu9.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu9.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster1.dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.dcache-maintenance\_latency**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.dcache-miss\_latency**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.dcache-prefetch\_enabled**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

Type: `bool`

Default value: `false`

**subcluster1.dcache-read\_access\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.dcache-size**

L1 D-Cache size in bytes.

Type: uint32\_t

Default value: 0x10000

**subcluster1.dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.dcache-write\_access\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.ete.CLAIMTAGS**

Number of claim tags.

Type: uint8\_t

Default value: 4

**subcluster1.ete.ETE\_REVISION**

ETE revision: 0=ETEv1.0, 1=ETEv1.1, 2=ETEv1.2, 3=ETEv1.3.

Type: uint8\_t

Default value: 1

**subcluster1.ete.MAX\_INST\_PER\_Q**

Maximum limit for the number of instructions implied by a Q element.

Type: uint16\_t

Default value: 0x1

**subcluster1.ete.NumberOfRSPairs**

Number of resource selector pairs.

Type: uint8\_t

Default value: 8

**subcluster1.ete.PIDR\_CM0D**

TRCPIDR CM0D value.

Type: uint8\_t

Default value: 0

**subcluster1.ete.PIDR\_REVAND**

TRCPIDR REVAND value.

Type: `uint8_t`

Default value: 0

#### **`subcluster1.ete.PIDR_REVISION`**

TRCPIDR REVISION value.

Type: `uint8_t`

Default value: 0

#### **`subcluster1.ete.Q_CADENCE`**

Number of instruction blocks traced between two Q elements.

Type: `uint16_t`

Default value: 0x1

#### **`subcluster1.ete.RES0_STATEFUL`**

Whether **RES0** bits are stateful or **RAZ/WI**.

Type: `bool`

Default value: `false`

#### **`subcluster1.ete.RETSTACK`**

Return stack depth.

Type: `uint8_t`

Default value: 1

#### **`subcluster1.ete.REVISION`**

TRCIDR1 revision value.

Type: `uint8_t`

Default value: 1

#### **`subcluster1.ete.SIM_OVERFLOW_GRANULARITY`**

Number of instruction blocks in each granule, for simulated overflow.

Type: `uint32_t`

Default value: 0x64

#### **`subcluster1.ete.SIM_OVERFLOW_PERCENTAGE`**

Percentage of instruction blocks lost in each granule, for simulated overflow.



Type: `uint8_t`

Default value: 0

#### **`subcluster1.ete.SOURCE_ADDRESS`**

Allow generation of source address elements.

Type: `bool`

Default value: `false`

#### **`subcluster1.ete.TRACE_OUTPUT`**

File to which to write trace byte stream.

Type: `string`

Default value: N/A

#### **`subcluster1.ete.TRCRSRTA_FORCED_EXCEP`**

TRCRSR.TA value for a forcibly traced exception.

Type: `bool`

Default value: `false`

#### **`subcluster1.ete.TSMARK`**

Whether timestamp markers are supported.

Type: `bool`

Default value: `true`

#### **`subcluster1.force_mte_tag_access_razwi_and_ignore_tag_checks`**

Force MTE tag accesses to **RAZ/WI** and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

Type: `bool`

Default value: `false`

#### **`subcluster1.force_zero_PSTATE_PAN`**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

### **`subcluster1.force_zero_mpam_partid_and_pmg`**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAMO\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

Type: `uint8_t`

Default value: 0

### **`subcluster1.has_delayed_dbgreg`**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

### **`subcluster1.has_delayed_sysreg`**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

### **`subcluster1.has_enhanced_pan`**

Implements Armv8.7 Enhanced PAN feature (FEAT\_PAN3) values of this parameter are:- 1, feature is implemented if ARMv8.7 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 2

### **`subcluster1.has_ete`**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (-plugin or -P).

Type: `bool`

Default value: `false`

### **`subcluster1.has_large_va`**

Implement support for the extended 52-bit virtual addresses from ARMv8.2 (FEAT\_LVA). values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.2 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

#### **subcluster1.icache-hit\_latency**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

Type: bool

Default value: false

#### **subcluster1.icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to

the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.icache-size`**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: `0x10000`

#### **`subcluster1.instruction_tlb_size`**

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

Type: `uint32_t`

Default value: `0x0`

#### **`subcluster1.invalidate_code_cache_on_icache_cmo`**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: `uint8_t`

Default value: `0`

#### **`subcluster1.memory_tagging_support_level`**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

Type: `uint8_t`

Default value: `3`

#### **`subcluster1.pmu-num_counters`**

Number of PMU counters implemented.

Type: `uint8_t`

Default value: `6`

#### **`subcluster1.ptw_latency`**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.stage12_tlb_size`**

Number of stage1+2 tlb entries. Valid values are 0 or  $\geq 4$ .

Type: `uint32_t`

Default value: `0x80`

### **`subcluster1.tlb_latency`**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.tlbi_stall_enabled`**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: `bool`

Default value: `false`

### **`subcluster1.treat_PAC_as_NOP`**

Non-architecture parameter. Treat Pointer Authentication as **NOP**. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are **NOP**. Branch, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

Type: `bool`

Default value: `false`

### **`subcluster1.walk_cache_latency`**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

### 3.34 ARMC1NanoCT\_C1ProCT\_C1UltraCT

Defined in `LISA/ARMC1NanoCT_C1ProCT_C1UltraCT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
C1Nano r0p0	Preliminary support
C1Pro r0p0	Preliminary support
C1Ultra r0p0	Preliminary support

For an explanation of the quality levels, see [Quality level definitions](#).

#### Changes in 11.31.15

The following parameters were added:

- `num_acp`
- `subcluster0.has_delayed_dbgreg`
- `subcluster0.has_delayed_sysreg`
- `subcluster1.has_delayed_dbgreg`
- `subcluster1.has_delayed_sysreg`
- `subcluster2.has_delayed_dbgreg`
- `subcluster2.has_delayed_sysreg`
- `subcluster2.mpam_has_altsp`

#### About ARMC1NanoCT\_C1ProCT\_C1UltraCT

The number of cores in each subcluster is configurable using the following parameters:

##### **subcluster0.NUM\_CORES**

Possible values are 1-12 (ARMC1NanoCT).

##### **subcluster1.NUM\_CORES**

Possible values are 1-12 (ARMC1ProCT).

##### **subcluster2.NUM\_CORES**

Possible values are 1-12 (ARMC1UltraCT).

The total number of cores in the cluster cannot exceed 14.

Port arrays are expanded to the width needed for the maximum number of cores in each subcluster. Use the following port array indexes:

- `<port_name>[0-11]` for cores in `subcluster0`.
- `<port_name>[12-23]` for cores in `subcluster1`.
- `<port_name>[24-35]` for cores in `subcluster2`.



All instances in the Master cross trigger matrix port array, `cti[36]` must be connected, regardless of the `NUM_CORES` values used.

Core-specific parameters have the following prefixes:

- `subcluster0.cpu0` to `subcluster0.cpu11` identify cores in `subcluster0`.
- `subcluster1.cpu0` to `subcluster1.cpu11` identify cores in `subcluster1`.
- `subcluster2.cpu0` to `subcluster2.cpu11` identify cores in `subcluster2`.

For information about the cores in this model, see:

- [ARMC1NanoCT](#).
- [ARMC1ProCT](#).
- [ARMC1UltraCT](#).

### Iris and MTI instances for ARMC1NanoCT\_C1ProCT\_C1UltraCT

This model has the following Iris instances:

Name	Instance type
ARMC1NanoCT_C1ProCT_C1UltraCT	Cluster_ARM_C1-Nano_C1-PRO_C1-ULTRA_Heterogeneous
ARMC1NanoCT_C1ProCT_C1UltraCT.AMU	PVBusLogger
ARMC1NanoCT_C1ProCT_C1UltraCT.AMU.mapper	PVBusMapper
ARMC1NanoCT_C1ProCT_C1UltraCT.DAP	PVBusLogger
ARMC1NanoCT_C1ProCT_C1UltraCT.DAP.mapper	PVBusMapper
ARMC1NanoCT_C1ProCT_C1UltraCT.DSU	C1-DSU
ARMC1NanoCT_C1ProCT_C1UltraCT.DSU.PPU_cluster	PPUv1
ARMC1NanoCT_C1ProCT_C1UltraCT.DSU.PPU_cluster.busslave	PVBusSlave
ARMC1NanoCT_C1ProCT_C1UltraCT.DSU.PPU_coreU (where $U = 0-2$ )	PPUv1
ARMC1NanoCT_C1ProCT_C1UltraCT.DSU.PPU_coreU.busslave (where $U = 0-2$ )	PVBusSlave
ARMC1NanoCT_C1ProCT_C1UltraCT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMC1NanoCT_C1ProCT_C1UltraCT.DSU.mpam_busslave	PVBusSlave
ARMC1NanoCT_C1ProCT_C1UltraCT.DSU.shared_cache	PVCache
ARMC1NanoCT_C1ProCT_C1UltraCT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMC1NanoCT_C1ProCT_C1UltraCT.DSU.shared_cache.upstream[U] (where $U = 0-8$ )	PVBusSlave
ARMC1NanoCT_C1ProCT_C1UltraCT.DSU.utility_slave[0]	PVBusSlave
ARMC1NanoCT_C1ProCT_C1UltraCT.MMAP	PVBusLogger
ARMC1NanoCT_C1ProCT_C1UltraCT.MMAP.mapper	PVBusMapper
ARMC1NanoCT_C1ProCT_C1UltraCT.RAS	PVBusLogger
ARMC1NanoCT_C1ProCT_C1UltraCT.RAS.mapper	PVBusMapper

Name	Instance type
ARMC1NanoCT_C1ProCT_C1UltraCT.cpuU.debug_rom (where $U = 0-2$ )	debug_rom
ARMC1NanoCT_C1ProCT_C1UltraCT.ext_bus	PVBusLogger
ARMC1NanoCT_C1ProCT_C1UltraCT.ext_bus.mapper	PVBusMapper
ARMC1NanoCT_C1ProCT_C1UltraCT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMC1NanoCT_C1ProCT_C1UltraCT.global_debug_rom	debug_rom
ARMC1NanoCT_C1ProCT_C1UltraCT.secondary_debug_rom	debug_rom
ARMC1NanoCT_C1ProCT_C1UltraCT.subcluster0	Subcluster_ARM_C1-Nano
ARMC1NanoCT_C1ProCT_C1UltraCT.subcluster0.cpu0	ARM_C1-Nano
ARMC1NanoCT_C1ProCT_C1UltraCT.subclusterU.cpu0.UTLB (where $U = 0-2$ )	TLB
ARMC1NanoCT_C1ProCT_C1UltraCT.subclusterU.cpuV.dtlb (where $U = 0-2$ ; $V = 0-2$ )	TLB
ARMC1NanoCT_C1ProCT_C1UltraCT.subclusterU.cpu0.l1dcache (where $U = 0-2$ )	PVCache
ARMC1NanoCT_C1ProCT_C1UltraCT.subclusterU.cpu0.l1dcache.downstream[0].pvbusmaster (where $U = 0-2$ )	PVBusMaster
ARMC1NanoCT_C1ProCT_C1UltraCT.subclusterU.cpu0.l1dcache.upstream[0] (where $U = 0-2$ )	PVBusSlave
ARMC1NanoCT_C1ProCT_C1UltraCT.subclusterU.cpu0.l1icache (where $U = 0-2$ )	PVCache
ARMC1NanoCT_C1ProCT_C1UltraCT.subclusterU.cpu0.l1icache.downstream[0].pvbusmaster (where $U = 0-2$ )	PVBusMaster
ARMC1NanoCT_C1ProCT_C1UltraCT.subclusterU.cpu0.l1icache.upstream[0] (where $U = 0-2$ )	PVBusSlave
ARMC1NanoCT_C1ProCT_C1UltraCT.subclusterU.cpu0.l2cache (where $U = 0-2$ )	PVCache
ARMC1NanoCT_C1ProCT_C1UltraCT.subclusterU.cpu0.l2cache.downstream[0].pvbusmaster (where $U = 0-2$ )	PVBusMaster
ARMC1NanoCT_C1ProCT_C1UltraCT.subclusterU.cpu0.l2cache.upstream[A] (where $U = 0-2$ ; $A = 0-1$ )	PVBusSlave
ARMC1NanoCT_C1ProCT_C1UltraCT.subclusterU.sve (where $U = 0-2$ )	ScalableVectorExtension
ARMC1NanoCT_C1ProCT_C1UltraCT.subcluster1	Subcluster_ARM_C1-Pro
ARMC1NanoCT_C1ProCT_C1UltraCT.subcluster1.cpu0	ARM_C1-Pro
ARMC1NanoCT_C1ProCT_C1UltraCT.subcluster2	Subcluster_ARM_C1-Ultra
ARMC1NanoCT_C1ProCT_C1UltraCT.subcluster2.cpu0	ARM_C1-Ultra

This model has the following MTI trace components:

Name	Component type
ARMC1NanoCT_C1ProCT_C1UltraCT.AMU	PVBusLogger
ARMC1NanoCT_C1ProCT_C1UltraCT.AMU.mapper	PVBusMapper
ARMC1NanoCT_C1ProCT_C1UltraCT.DAP	PVBusLogger
ARMC1NanoCT_C1ProCT_C1UltraCT.DAP.mapper	PVBusMapper
ARMC1NanoCT_C1ProCT_C1UltraCT.DSU	C1-DSU
ARMC1NanoCT_C1ProCT_C1UltraCT.DSU.PPU_cluster	PPUv1
ARMC1NanoCT_C1ProCT_C1UltraCT.DSU.PPU_cluster.busslave	PVBusSlave



Name	Component type
ARMC1NanoCT_C1ProCT_C1UltraCT.DSU.PPU_coreU (where $U = 0-2$ )	PPUv1
ARMC1NanoCT_C1ProCT_C1UltraCT.DSU.PPU_coreU.busslave (where $U = 0-2$ )	PVBusSlave
ARMC1NanoCT_C1ProCT_C1UltraCT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMC1NanoCT_C1ProCT_C1UltraCT.DSU.mpam_busslave	PVBusSlave
ARMC1NanoCT_C1ProCT_C1UltraCT.DSU.shared_cache	PVCache
ARMC1NanoCT_C1ProCT_C1UltraCT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMC1NanoCT_C1ProCT_C1UltraCT.DSU.shared_cache.upstream[U] (where $U = 0-8$ )	PVBusSlave
ARMC1NanoCT_C1ProCT_C1UltraCT.DSU.utility_slave[0]	PVBusSlave
ARMC1NanoCT_C1ProCT_C1UltraCT.MMAP	PVBusLogger
ARMC1NanoCT_C1ProCT_C1UltraCT.MMAP.mapper	PVBusMapper
ARMC1NanoCT_C1ProCT_C1UltraCT.RAS	PVBusLogger
ARMC1NanoCT_C1ProCT_C1UltraCT.RAS.mapper	PVBusMapper
ARMC1NanoCT_C1ProCT_C1UltraCT.ext_bus	PVBusLogger
ARMC1NanoCT_C1ProCT_C1UltraCT.ext_bus.mapper	PVBusMapper
ARMC1NanoCT_C1ProCT_C1UltraCT.gic_cpuif_decoder_cluster	GlCv3CPUInterfaceDecoder
ARMC1NanoCT_C1ProCT_C1UltraCT.subcluster0.cpu0	ARM_C1-Nano
ARMC1NanoCT_C1ProCT_C1UltraCT.subclusterU.cpu0.UTLB (where $U = 0-2$ )	TLB
ARMC1NanoCT_C1ProCT_C1UltraCT.subclusterU.cpu0.l1dcache (where $U = 0-2$ )	PVCache
ARMC1NanoCT_C1ProCT_C1UltraCT.subclusterU.cpu0.l1dcache.downstream[0].pvbusmaster (where $U = 0-2$ )	PVBusMaster
ARMC1NanoCT_C1ProCT_C1UltraCT.subclusterU.cpu0.l1dcache.upstream[0] (where $U = 0-2$ )	PVBusSlave
ARMC1NanoCT_C1ProCT_C1UltraCT.subclusterU.cpu0.l1icache (where $U = 0-2$ )	PVCache
ARMC1NanoCT_C1ProCT_C1UltraCT.subclusterU.cpu0.l1icache.downstream[0].pvbusmaster (where $U = 0-2$ )	PVBusMaster
ARMC1NanoCT_C1ProCT_C1UltraCT.subclusterU.cpu0.l1icache.upstream[0] (where $U = 0-2$ )	PVBusSlave
ARMC1NanoCT_C1ProCT_C1UltraCT.subclusterU.cpu0.l2cache (where $U = 0-2$ )	PVCache
ARMC1NanoCT_C1ProCT_C1UltraCT.subclusterU.cpu0.l2cache.downstream[0].pvbusmaster (where $U = 0-2$ )	PVBusMaster
ARMC1NanoCT_C1ProCT_C1UltraCT.subclusterU.cpu0.l2cache.upstream[A] (where $U = 0-2$ ; $A = 0-1$ )	PVBusSlave
ARMC1NanoCT_C1ProCT_C1UltraCT.subcluster1.cpu0	ARM_C1-Pro
ARMC1NanoCT_C1ProCT_C1UltraCT.subcluster2.cpu0	ARM_C1-Ultra

### Ports for ARMC1NanoCT\_C1ProCT\_C1UltraCT

Port	Direction	Protocol	Description
acp_s	slave	PVBus	AXI ACP subordinate port.
AENDOMP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).

Port	Direction	Protocol	Description
AEND2MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND3MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART2MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	slave	Signal	CHI defined pins.
broadcastcachemaint	slave	Signal	ACE defined pins.
broadcastouter	slave	Signal	ACE defined pins.
broadcastpersist	slave	Signal	CHI defined pins.
cfgend	slave	Signal	This signal is for EE bit initialisation.
cfgsdisable	slave	Signal	This signal disables write access to some secure Interrupt Controller registers.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcsmp_channel	master	PChannel	Cluster PCSM signal
cluster_powerdown_out	master	Signal	DEPRECATED - An output from PPU that informs thermal controller of the cluster power info
cluster_ppu_hw_stat	master	Value	Cluster PPU power state
clusterid	slave	Value	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set ManagerID64 of the core.
clusterpmuirq	master	Signal	DynamiQ pmu irq
cme_pcsmp_channel	master	PChannel	Cluster PCSM signal
cmeerirq	master	Signal	RAS cme err irq
cmefaultirq	master	Signal	RAS cme fault irq
CNTHPIRQ	master	Signal	Timer signals to SOC.
CNTHPSIRQ	master	Signal	Timer signals to SOC
CNTHVIRQ	master	Signal	Timer signals to SOC.
CNTHVSIRQ	master	Signal	Timer signals to SOC

Port	Direction	Protocol	Description
CNTPNSIRQ	master	Signal	Timer signals to SOC.
CNTPSIRQ	master	Signal	Timer signals to SOC.
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.
CNTVIRQ	master	Signal	Timer signals to SOC.
commirq	master	Signal	Interrupt signal from debug communications channel.
core_clk_in	slave	ClockSignal	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
core_pcsn_pchannel	master	PChannel	Core PCSM signals
core_powerdown_out	master	Signal	DEPRECATED - An output from PPU that informs thermal controller of the core power info
core_ppu_hw_stat	master	Value	Notify the power state change inside the PPU.
coreerrirq	master	Signal	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq	master	Signal	Fault indicator for a detected 1 or 2 bit ECC error
coreinstrrun	master	Signal	Core Running State
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
ctidbgirq	master	Signal	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	slave	Signal	External debug interface.
dbgnopwrdown	master	Signal	No power-down request.
dbgpwrupreq	master	Signal	Debug power up request.
defaultMP	slave	Signal	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main manager interface ports.
dev_debug_s	slave	PVBus	External debug interface.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.
gicreset	master	Signal	An output from PPU that can be used to reset GICCLK domain components
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	slave	PVBus	External debug interface.
pmbirq	master	Signal	Interrupt signal from the statistical profiling unit.
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	master	Signal	PPU cluster interrupt.
ppu_cluster_isolate	master	Signal	PPU Cluster Domain isolation control.
ppu_cluster_wakerequest	slave	Signal	PPU cluster wake request signal.
ppu_cme_irq	master	Signal	PPU CME interrupt
ppu_core_irq	master	Signal	PPU core interrupt.

Port	Direction	Protocol	Description
ppu_core_isolate	master	Signal	PPU Core Domain isolation control.
ppu_core_wakerequest	slave	Signal	PPU core wake request signal.
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbush_m0	master	PVBus	The core will generate bus requests on this port.
pvbush_periph_m	master	PVBus	The core can generate peripheral bus request on this port.
rei	slave	Signal	Per core RAM Error Interrupt.
reset	slave	Signal	Reset signal to cluster.
rvbaraddr	slave	Value_64	Reset vector base address.
sci_m	master	SystemCoherencyInterface	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei	slave	Signal	Per core System Error physical pins.
spiden	slave	Signal	External debug interface.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq	master	Signal	Interrupt signal from the trace buffer unit.
utility_bus_s	slave	PVBus	Utility bus subordinate
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfiq	slave	Signal	Virtualised FIQ.
virq	slave	Signal	Virtualised IRQ.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Per core virtual System Error physical pins.

## Parameters for ARMC1NanoCT\_C1ProCT\_C1UltraCT

### AEND0\_DEFAULT

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

Type: uint64\_t

Default value: 0x0

### AEND1\_DEFAULT

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

Type: uint64\_t

Default value: 0x0

**AEND2\_DEFAULT**

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

Type: `uint64_t`

Default value: `0x0`

**AEND3\_DEFAULT**

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART0\_DEFAULT**

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART1\_DEFAULT**

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART2\_DEFAULT**

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART3\_DEFAULT**

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

Type: `uint64_t`

Default value: `0x0`

**BROADCASTATOMIC**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`

Default value: `true`

**BROADCASTCACHEMAINT**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

Type: `bool`

Default value: `false`

**BROADCASTOUTER**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

Type: `bool`

Default value: `false`

**BROADCASTPERSIST**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: `bool`

Default value: `true`

**CLUSTER\_ID**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the ManagerID64 of the core.

Type: `uint32_t`

Default value: `0x0`

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. This value may affect performance of the simulation, but

is required if the rest of the system is sensitive to certain even when cache state modelling is disabled. Possible values are:- 0 = All CMOs are broadcast if architecturally required- 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required.

Type: `uint8_t`

Default value: 1

### **DBGROMADDR**

Initialization value of DBGDRAR register. Bits[55:12] of this register specify the ROM table physical address.

Type: `uint64_t`

Default value: 0x0

### **DBGROMADDRV**

If true, set bits[1:0] of the CP15 DBGDRAR to indicate that the address is valid.

Type: `bool`

Default value: `false`

### **GICDISABLE**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`

Default value: `true`

### **NUM\_CMES**

Defines how many CMEs are associated with each cluster. This parameter must be in sync with SVE.ScalableVectorExtension SME configuration.

Type: `uint8_t`

Default value: 0

### **bus\_type**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

Type: `uint8_t`

Default value: 0

**core\_complex\_mapping**

Defines Complex descriptions for platforms that support several Cores per Complex like Cortex-A510. JSON format: {"complex0": { "cores" : [ 0, 1 ], "l2-cache" :{"exists":1, "size":16MB}}, ... , "complexN": { "cores" : [<core\_list>], "l2-cache" : {"exists":1, "size":16MB}}} where <core\_list> is the list of cores in the complexN. Effective only when the parameter value is not empty.

Type: string

Default value: "{ \"complex0\": { \"cores\": [0, 1], \"l2-cache\" :{ \"exists\":1, \"size \": \"16MB\" }}, \"complex1\": { \"cores\": [2, 3], \"l2-cache\" :{ \"exists\":1, \"size \": \"16MB\" }}, \"complex2\": { \"cores\": [4, 5], \"l2-cache\" :{ \"exists\":1, \"size \": \"16MB\" }}, \"complex3\": { \"cores\": [6, 7], \"l2-cache\" :{ \"exists\":1, \"size \": \"16MB\" }}, \"complex4\": { \"cores\": [8, 9], \"l2-cache\" :{ \"exists\":1, \"size \": \"16MB\" }}, \"complex5\": { \"cores\": [10, 11], \"l2-cache\" :{ \"exists\":1, \"size \": \"16MB\" }}, \"complex6\": { \"cores\": [12, 13], \"l2-cache\" :{ \"exists\":1, \"size \": \"16MB\" } } }

**core\_power\_on\_by\_default**

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPUs which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

Type: bool

Default value: false

**dcache-state\_modelled**

Set whether D-cache has stateful implementation.

Type: bool

Default value: false

**enable\_simulation\_performance\_optimizations**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

Type: bool

Default value: true

**has\_peripheral\_port**

If true, an additional AXI peripheral port is configured.

Type: bool



Default value: `false`

### **`icache-state_modelled`**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **`l3cache-has_mpam`**

L3 Cache has MPAM support.

Type: `bool`

Default value: `true`

### **`l3cache-hit_latency`**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-maintenance_latency`**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-miss_latency`**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-mpamf.max_partid_ns`**

Maximum value of non-secure PARTID supported for DSU L3Cache. Options are 7 or 63.

Type: `uint8_t`

Default value: 63

**l3cache-read\_access\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size l3cache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**l3cache-read\_bus\_width\_in\_bytes**

L3 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**l3cache-read\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. l3cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**l3cache-size**

L3 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**l3cache-snoop\_data\_transfer\_latency**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**l3cache-snoop\_issue\_latency**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **l3cache-ways**

L3 Cache number of ways (sets are implicit from size).

Type: `uint16_t`

Default value: 16

### **l3cache-write\_access\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size `l3cache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **l3cache-write\_bus\_width\_in\_bytes**

L3 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: `uint32_t`

Default value: 0x10

### **l3cache-write\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l3cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **mpmm\_accumulator\_multiplier**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator. value of `n` means the accumulator will use (`n * accumulator value`) to calculate the mpmm threshold (MPMM). is provided as a fast model workaround to handle cases where execution of quantums in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

Type: `uint8_t`

Default value: 1

### **num\_acp**

Number of ACP ports.

Type: `uint8_t`

Default value: 0

### **num\_nodes**

Number of transport nodes. Zero implies direct-connect configuration.

Type: `uint8_t`

Default value: 1

### **scu\_cache\_protection**

SCU-L3 is configured with ECC if true.

Type: `bool`

Default value: `true`

### **subcluster0.CPUCFR**

Value of CPU Configuration Register.

Type: `uint64_t`

Default value: `0x0`

### **subcluster0.NUM\_CORES**

Number of cores in the subcluster. Total number of cores in cluster may not exceed 14.

Type: `uint8_t`

Default value: 1

### **subcluster0.core\_cache\_protection**

`core_cache_protection` can change `ERROFR`, `ERROPFGF` and `ERROPFGCTL` fields. Possible values are: -1:Not implemented (by default), 0:Disabled, 1:Enabled.

Type: `int8_t`

Default value: 1

### **subcluster0.cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**subcluster0.cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**subcluster0.cpu0.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster0.cpu0.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster0.cpu0.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu0.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu0.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu0.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu0.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu0.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu0.l2cache-read_bus_width_in_bytes`**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: `uint32_t`

Default value: `0x10`

### **`subcluster0.cpu0.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu0.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

**subcluster0.cpu0.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu0.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu0.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu0.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu0.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu0.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu0.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu0.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster0.cpu0.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t



Default value: 171

**subcluster0.cpu0.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu0.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu0.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu0.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu0.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu0.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu0.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: 0xf000000

**subcluster0.cpu0.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu0.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster0.cpu0.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: true

**subcluster0.cpu1.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: false

**subcluster0.cpu1.CRYPTODISABLE**

Disable cryptographic features.

Type: bool

Default value: false

**subcluster0.cpu1.RVBARADDR**

Value of RVBAR\_ELx register.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter trace\_special\_hlt\_imm16.

Type: bool

Default value: false

**subcluster0.cpu1.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu1.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu1.l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

**subcluster0.cpu1.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu1.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu1.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: `uint8_t`

Default value: `8`

**subcluster0.cpu1.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu1.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu1.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu1.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu1.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu1.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu1.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

### **`subcluster0.cpu1.semihosting-T32_HLT`**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

### **`subcluster0.cpu1.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

### **`subcluster0.cpu1.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

### **`subcluster0.cpu1.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

### **`subcluster0.cpu1.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster0.cpu1.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu1.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf0000000`

### **`subcluster0.cpu1.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster0.cpu1.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf0000000`

### **`subcluster0.cpu1.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu1.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu1.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu10.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu10.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu10.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu10.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu10.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu10.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu10.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`



**subcluster0.cpu10.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu10.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu10.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu10.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu10.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu10.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu10.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

#### **subcluster0.cpu10.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu10.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

#### **subcluster0.cpu10.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu10.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

#### **subcluster0.cpu10.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu10.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu10.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu10.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu10.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster0.cpu10.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster0.cpu10.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu10.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu10.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: true

**subcluster0.cpu10.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu10.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu10.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster0.cpu10.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu10.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu10.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu10.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu11.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu11.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu11.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu11.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu11.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu11.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu11.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu11.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu11.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: `uint32_t`

Default value: `0x10`

**subcluster0.cpu11.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu11.l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: 0x80000

**subcluster0.cpu11.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu11.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu11.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu11.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu11.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu11.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu11.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu11.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu11.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu11.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu11.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu11.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster0.cpu11.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.



Type: `uint8_t`

Default value: 171

### **`subcluster0.cpu11.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **`subcluster0.cpu11.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **`subcluster0.cpu11.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster0.cpu11.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu11.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu11.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster0.cpu11.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

#### **`subcluster0.cpu11.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster0.cpu11.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu11.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster0.cpu2.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu2.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu2.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: 0x0

**subcluster0.cpu2.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter trace\_special\_hlt\_imm16.

Type: bool

Default value: false

**subcluster0.cpu2.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

#### **subcluster0.cpu2.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu2.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

#### **subcluster0.cpu2.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu2.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu2.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

#### **subcluster0.cpu2.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu2.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu2.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu2.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu2.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu2.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu2.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster0.cpu2.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster0.cpu2.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu2.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu2.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu2.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu2.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu2.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu2.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu2.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu2.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster0.cpu2.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster0.cpu3.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster0.cpu3.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster0.cpu3.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu3.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu3.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu3.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`



**subcluster0.cpu3.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu3.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu3.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu3.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu3.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu3.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu3.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu3.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu3.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu3.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster0.cpu3.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster0.cpu3.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster0.cpu3.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster0.cpu3.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu3.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu3.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu3.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu3.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu3.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster0.cpu3.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu3.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster0.cpu4.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu4.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu4.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu4.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu4.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu4.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu4.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu4.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu4.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu4.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu4.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu4.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu4.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu4.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.



Type: `uint8_t`

Default value: 60

### **subcluster0.cpu4.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

### **subcluster0.cpu4.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **subcluster0.cpu4.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **subcluster0.cpu4.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **subcluster0.cpu4.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **subcluster0.cpu4.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **subcluster0.cpu4.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

#### **`subcluster0.cpu4.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

#### **`subcluster0.cpu4.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster0.cpu4.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu4.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster0.cpu5.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu5.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu5.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu5.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu5.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu5.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu5.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu5.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster0.cpu5.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu5.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu5.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu5.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu5.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu5.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu5.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu5.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu5.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu5.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster0.cpu5.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster0.cpu5.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu5.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu5.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu5.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu5.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster0.cpu5.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu5.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu5.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster0.cpu5.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu6.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu6.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu6.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu6.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu6.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu6.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`



**subcluster0.cpu6.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu6.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu6.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu6.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu6.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu6.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu6.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu6.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu6.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu6.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster0.cpu6.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster0.cpu6.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster0.cpu6.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster0.cpu6.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu6.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu6.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu6.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu6.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu6.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster0.cpu6.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu6.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster0.cpu7.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu7.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu7.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu7.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu7.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu7.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu7.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

#### **subcluster0.cpu7.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu7.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu7.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

#### **subcluster0.cpu7.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu7.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu7.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu7.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu7.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu7.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu7.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu7.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.



Type: `uint8_t`

Default value: 60

### **`subcluster0.cpu7.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

### **`subcluster0.cpu7.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **`subcluster0.cpu7.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **`subcluster0.cpu7.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster0.cpu7.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu7.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu7.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster0.cpu7.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu7.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu7.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu7.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu8.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu8.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu8.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu8.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu8.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu8.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu8.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu8.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster0.cpu8.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu8.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu8.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu8.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu8.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu8.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu8.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu8.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu8.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu8.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster0.cpu8.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster0.cpu8.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu8.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu8.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu8.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu8.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu8.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu8.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu8.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu8.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster0.cpu8.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu9.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu9.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu9.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu9.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu9.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu9.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`



**subcluster0.cpu9.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu9.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu9.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu9.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu9.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu9.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu9.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu9.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu9.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu9.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster0.cpu9.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster0.cpu9.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster0.cpu9.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster0.cpu9.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu9.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu9.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu9.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu9.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu9.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu9.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu9.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.dcache-hit_latency`**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.dcache-maintenance_latency`**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.dcache-miss_latency`**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.dcache-prefetch_enabled`**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

Type: `bool`

Default value: `false`

### **subcluster0.dcache-read\_access\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **subcluster0.dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **subcluster0.dcache-size**

L1 D-Cache size in bytes.

Type: `uint32_t`

Default value: `0x8000`

### **subcluster0.dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **subcluster0.dcache-write\_access\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.ete.CLAIMTAGS**

Number of claim tags.

Type: uint8\_t

Default value: 4

**subcluster0.ete.MAX\_INST\_PER\_Q**

Maximum limit for the number of instructions implied by a Q element.

Type: uint16\_t

Default value: 0x1

**subcluster0.ete.PIDR\_CM0D**

TRCPIDR CM0D value.

Type: uint8\_t

Default value: 0

**subcluster0.ete.PIDR\_REVAND**

TRCPIDR REVAND value.

Type: uint8\_t

Default value: 0

**subcluster0.ete.PIDR\_REVISION**

TRCPIDR REVISION value.

Type: uint8\_t

Default value: 0

**subcluster0.ete.Q\_CADENCE**

Number of instruction blocks traced between two Q elements.

Type: `uint16_t`

Default value: `0x1`

#### **`subcluster0.ete.RES0_STATEFUL`**

Whether **RES0** bits are stateful or **RAZ/WI**.

Type: `bool`

Default value: `false`

#### **`subcluster0.ete.RETSTACK`**

Return stack depth.

Type: `uint8_t`

Default value: `3`

#### **`subcluster0.ete.REVISION`**

TRCIDR1 revision value.

Type: `uint8_t`

Default value: `0`

#### **`subcluster0.ete.SIM_OVERFLOW_GRANULARITY`**

Number of instruction blocks in each granule, for simulated overflow.

Type: `uint32_t`

Default value: `0x64`

#### **`subcluster0.ete.SIM_OVERFLOW_PERCENTAGE`**

Percentage of instruction blocks lost in each granule, for simulated overflow.

Type: `uint8_t`

Default value: `0`

#### **`subcluster0.ete.SOURCE_ADDRESS`**

Allow generation of source address elements.

Type: `bool`

Default value: `false`

#### **`subcluster0.ete.TRACE_OUTPUT`**

File to which to write trace byte stream.



Type: `string`

Default value: `N/A`

#### **`subcluster0.ete.TRCRSRTA_FORCED_EXCEP`**

TRCRSR.TA value for a forcibly traced exception.

Type: `bool`

Default value: `false`

#### **`subcluster0.force_mte_tag_access_razwi_and_ignore_tag_checks`**

Force MTE tag accesses to **RAZ/WI** and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

Type: `bool`

Default value: `false`

#### **`subcluster0.force_zero_PSTATE_PAN`**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

#### **`subcluster0.force_zero_mpam_partid_and_pmg`**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

Type: `uint8_t`

Default value: `0`

#### **`subcluster0.has_actlr2`**

If true ACLTR2 exists and ACTLR2(NS) is aliased to ACTLR\_EL1[63:32].

Type: `bool`

Default value: `true`

#### **`subcluster0.has_delayed_dbgreg`**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

### **`subcluster0.has_delayed_sysreg`**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

### **`subcluster0.has_dot_product`**

Possible values of this parameter are:- 1, feature not implemented.- 2, feature is implemented.

Type: `uint8_t`

Default value: 2

### **`subcluster0.has_ete`**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (-plugin or -P).

Type: `bool`

Default value: `false`

### **`subcluster0.icache-hit_latency`**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **`subcluster0.icache-maintenance_latency`**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **`subcluster0.icache-miss_latency`**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **subcluster0.icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

Type: bool

Default value: false

### **subcluster0.icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster0.icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster0.icache-size**

L1 I-Cache size in bytes.

Type: uint32\_t

Default value: 0x8000

### **subcluster0.invalidate\_code\_cache\_on\_icache\_cmo**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: uint8\_t

Default value: 0

**subcluster0.memory\_tagging\_support\_level**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

Type: `uint8_t`

Default value: 3

**subcluster0.pmu-num\_counters**

Number of PMU counters implemented.

Type: `uint8_t`

Default value: 6

**subcluster0.ptw\_latency**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

**subcluster0.ras\_pfg\_clock\_mhz**

RAS Pseudo-Fault generation clock rate in MHz.

Type: `uint8_t`

Default value: 12

**subcluster0.tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

**subcluster0.tlbi\_stall\_enabled**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: `bool`

Default value: `false`

**subcluster0.treat\_PAC\_as\_NOP**

Non-architecture parameter. Treat Pointer Authentication as **NOP**. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are **NOP**. Branch, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

Type: `bool`

Default value: `false`

**subcluster0.walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.CPUCFR**

Value of CPU Configuration Register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.NUM\_CORES**

Number of cores in the subcluster. Total number of cores in cluster may not exceed 14.

Type: `uint8_t`

Default value: `1`

**subcluster1.core\_cache\_protection**

`core_cache_protection` can change `ERROFR`, `ERROPFGF` and `ERROPFGCTL` fields. Possible values are: -1:Not implemented (by default), 0:Disabled, 1:Enabled.

Type: `int8_t`

Default value: `1`

**subcluster1.cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**subcluster1.cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**subcluster1.cpu0.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu0.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster1.cpu0.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu0.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu0.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu0.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

#### **`subcluster1.cpu0.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu0.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu0.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu0.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu0.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu0.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x100000`

### **`subcluster1.cpu0.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu0.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu0.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu0.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu0.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

### **`subcluster1.cpu0.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).



Type: `uint8_t`

Default value: 0

#### **`subcluster1.cpu0.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

#### **`subcluster1.cpu0.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

#### **`subcluster1.cpu0.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: 0x123456

#### **`subcluster1.cpu0.semihosting-T32_HLT`**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

#### **`subcluster1.cpu0.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

#### **`subcluster1.cpu0.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

#### **`subcluster1.cpu0.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

### **`subcluster1.cpu0.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster1.cpu0.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu0.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster1.cpu0.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster1.cpu0.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster1.cpu0.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu0.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu0.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster1.cpu1.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu1.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster1.cpu1.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu1.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu1.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu1.force-fpsid`**

Override the FPSID value.

Type: `bool`

Default value: `true`

### **`subcluster1.cpu1.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to

the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu1.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x100000`

#### **`subcluster1.cpu1.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu1.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu1.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu1.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu1.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu1.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu1.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu1.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu1.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu1.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster1.cpu1.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster1.cpu1.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu1.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu1.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu1.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu1.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu1.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu1.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: 0xf000000

**subcluster1.cpu1.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu1.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster1.cpu1.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: true

**subcluster1.cpu10.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: false

**subcluster1.cpu10.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: bool

Default value: false

**subcluster1.cpu10.CRYPTODISABLE**

Disable cryptographic features.

Type: bool

Default value: false



**subcluster1.cpu10.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu10.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu10.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu10.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu10.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu10.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu10.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu10.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu10.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x100000

**subcluster1.cpu10.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu10.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu10.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu10.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu10.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

### **`subcluster1.cpu10.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

### **`subcluster1.cpu10.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu10.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu10.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu10.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster1.cpu10.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster1.cpu10.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster1.cpu10.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster1.cpu10.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu10.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu10.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu10.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu10.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu10.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu10.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu10.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster1.cpu11.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu11.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster1.cpu11.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu11.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu11.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu11.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu11.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu11.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu11.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu11.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu11.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu11.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x100000`

### **`subcluster1.cpu11.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu11.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu11.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu11.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu11.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu11.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu11.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000



**subcluster1.cpu11.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu11.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu11.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster1.cpu11.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster1.cpu11.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu11.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu11.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu11.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu11.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu11.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu11.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu11.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu11.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu11.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster1.cpu2.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster1.cpu2.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu2.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu2.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu2.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu2.force-fpsid`**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu2.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: 0x100000

**subcluster1.cpu2.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu2.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu2.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu2.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu2.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu2.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu2.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu2.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu2.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu2.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster1.cpu2.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster1.cpu2.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster1.cpu2.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster1.cpu2.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: true

**subcluster1.cpu2.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu2.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu2.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster1.cpu2.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu2.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu2.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu2.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster1.cpu3.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu3.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster1.cpu3.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu3.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.



Type: `bool`

Default value: `false`

### **`subcluster1.cpu3.force-fpsid`**

Override the FPSID value.

Type: `bool`

Default value: `true`

### **`subcluster1.cpu3.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu3.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu3.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu3.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu3.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to

the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu3.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x100000`

#### **`subcluster1.cpu3.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu3.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu3.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu3.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu3.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu3.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu3.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu3.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu3.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster1.cpu3.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster1.cpu3.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu3.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu3.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu3.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu3.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu3.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: 0xf000000

#### **subcluster1.cpu3.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

#### **subcluster1.cpu3.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

#### **subcluster1.cpu3.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: true

#### **subcluster1.cpu4.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: false

#### **subcluster1.cpu4.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: bool

Default value: false

#### **subcluster1.cpu4.CRYPTODISABLE**

Disable cryptographic features.

Type: bool

Default value: false

**subcluster1.cpu4.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu4.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu4.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu4.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu4.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu4.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu4.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu4.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu4.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x100000

**subcluster1.cpu4.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu4.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu4.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu4.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu4.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

### **`subcluster1.cpu4.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

### **`subcluster1.cpu4.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu4.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu4.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`



**subcluster1.cpu4.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster1.cpu4.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster1.cpu4.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster1.cpu4.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster1.cpu4.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu4.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu4.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu4.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu4.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu4.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu4.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu4.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster1.cpu5.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu5.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster1.cpu5.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu5.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu5.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu5.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu5.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu5.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu5.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu5.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu5.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu5.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x100000`

### **`subcluster1.cpu5.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu5.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu5.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu5.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu5.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu5.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu5.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu5.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu5.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu5.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster1.cpu5.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster1.cpu5.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu5.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu5.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu5.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu5.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu5.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster1.cpu5.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu5.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu5.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster1.cpu5.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster1.cpu6.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu6.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu6.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu6.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu6.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu6.force-fpsid`**

Override the FPSID value.

Type: `bool`

Default value: `true`



**subcluster1.cpu6.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu6.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu6.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu6.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu6.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu6.l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: 0x100000

**subcluster1.cpu6.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu6.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu6.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu6.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu6.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu6.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu6.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster1.cpu6.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster1.cpu6.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: 0x123456

**subcluster1.cpu6.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster1.cpu6.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster1.cpu6.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster1.cpu6.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster1.cpu6.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: true

**subcluster1.cpu6.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu6.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu6.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster1.cpu6.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu6.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu6.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu6.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster1.cpu7.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu7.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster1.cpu7.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu7.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu7.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu7.force-fpsid`**

Override the FPSID value.

Type: `bool`

Default value: `true`

### **`subcluster1.cpu7.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu7.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu7.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu7.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu7.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to

the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu7.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x100000`

#### **`subcluster1.cpu7.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu7.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu7.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu7.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu7.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu7.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu7.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu7.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu7.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu7.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster1.cpu7.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t



Default value: 171

**subcluster1.cpu7.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu7.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu7.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu7.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu7.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu7.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu7.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: 0xf000000

**subcluster1.cpu7.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu7.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster1.cpu7.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: true

**subcluster1.cpu8.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: false

**subcluster1.cpu8.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: bool

Default value: false

**subcluster1.cpu8.CRYPTODISABLE**

Disable cryptographic features.

Type: bool

Default value: false

**subcluster1.cpu8.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu8.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu8.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu8.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu8.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu8.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu8.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu8.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu8.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x100000

**subcluster1.cpu8.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu8.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu8.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu8.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu8.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

### **`subcluster1.cpu8.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

### **`subcluster1.cpu8.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu8.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu8.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu8.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster1.cpu8.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster1.cpu8.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster1.cpu8.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster1.cpu8.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu8.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu8.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu8.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu8.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu8.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu8.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu8.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster1.cpu9.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu9.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster1.cpu9.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu9.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu9.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu9.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu9.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu9.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.



Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu9.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu9.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu9.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu9.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x100000`

### **`subcluster1.cpu9.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu9.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu9.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu9.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu9.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu9.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu9.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster1.cpu9.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster1.cpu9.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu9.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu9.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu9.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu9.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster1.cpu9.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu9.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu9.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster1.cpu9.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster1.dcache-hit_latency`**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.dcache-maintenance_latency`**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.dcache-miss_latency`**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.dcache-prefetch_enabled`**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

Type: `bool`

Default value: `false`

### **`subcluster1.dcache-read_access_latency`**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.dcache-size**

L1 D-Cache size in bytes.

Type: `uint32_t`

Default value: `0x10000`

**subcluster1.dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.dcache-write\_access\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.etc.CLAIMTAGS**

Number of claim tags.

Type: `uint8_t`

Default value: 4

**subcluster1.ete.ETE\_REVISION**

ETE revision: 0=ETEv1.0, 1=ETEv1.1, 2=ETEv1.2, 3=ETEv1.3.

Type: uint8\_t

Default value: 1

**subcluster1.ete.MAX\_INST\_PER\_Q**

Maximum limit for the number of instructions implied by a Q element.

Type: uint16\_t

Default value: 0x1

**subcluster1.ete.NumberOfRSPairs**

Number of resource selector pairs.

Type: uint8\_t

Default value: 8

**subcluster1.ete.PIDR\_CM0D**

TRCPIDR CM0D value.

Type: uint8\_t

Default value: 0

**subcluster1.ete.PIDR\_REVAND**

TRCPIDR REVAND value.

Type: uint8\_t

Default value: 0

**subcluster1.ete.PIDR\_REVISION**

TRCPIDR REVISION value.

Type: uint8\_t

Default value: 0

**subcluster1.ete.Q\_CADENCE**

Number of instruction blocks traced between two Q elements.

Type: uint16\_t

Default value: 0x1

#### **subcluster1.ete.RES0\_STATEFUL**

Whether **RES0** bits are stateful or **RAZ/WI**.

Type: bool

Default value: false

#### **subcluster1.ete.RETSTACK**

Return stack depth.

Type: uint8\_t

Default value: 1

#### **subcluster1.ete.REVISION**

TRCIDR1 revision value.

Type: uint8\_t

Default value: 1

#### **subcluster1.ete.SIM\_OVERFLOW\_GRANULARITY**

Number of instruction blocks in each granule, for simulated overflow.

Type: uint32\_t

Default value: 0x64

#### **subcluster1.ete.SIM\_OVERFLOW\_PERCENTAGE**

Percentage of instruction blocks lost in each granule, for simulated overflow.

Type: uint8\_t

Default value: 0

#### **subcluster1.ete.SOURCE\_ADDRESS**

Allow generation of source address elements.

Type: bool

Default value: false

#### **subcluster1.ete.TRACE\_OUTPUT**

File to which to write trace byte stream.

Type: string



Default value: `N/A`

**subcluster1.ete.TRCRSRTA\_FORCED\_EXCEP**

TRCRSR.TA value for a forcibly traced exception.

Type: `bool`

Default value: `false`

**subcluster1.ete.TSMARK**

Whether timestamp markers are supported.

Type: `bool`

Default value: `true`

**subcluster1.force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks**

Force MTE tag accesses to **RAZ/WI** and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

Type: `bool`

Default value: `false`

**subcluster1.force\_zero\_PSTATE\_PAN**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

**subcluster1.force\_zero\_mpam\_partid\_and\_pmg**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAMO\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

Type: `uint8_t`

Default value: `0`

**subcluster1.has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

### **subcluster1.has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

### **subcluster1.has\_enhanced\_pan**

Implements Armv8.7 Enhanced PAN feature (FEAT\_PAN3) values of this parameter are:- 1, feature is implemented if ARMv8.7 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 2

### **subcluster1.has\_ete**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (-plugin or -P).

Type: `bool`

Default value: `false`

### **subcluster1.has\_large\_va**

Implement support for the extended 52-bit virtual addresses from ARMv8.2 (FEAT\_LVA). values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.2 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **subcluster1.icache-hit\_latency**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **subcluster1.icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

#### **subcluster1.icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

Type: bool

Default value: false

#### **subcluster1.icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.icache-size**

L1 I-Cache size in bytes.

Type: uint32\_t

Default value: 0x10000

#### **subcluster1.instruction\_tlb\_size**

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

Type: `uint32_t`

Default value: `0x0`

#### **`subcluster1.invalidate_code_cache_on_icache_cmo`**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: `uint8_t`

Default value: `0`

#### **`subcluster1.memory_tagging_support_level`**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

Type: `uint8_t`

Default value: `3`

#### **`subcluster1.pmu-num_counters`**

Number of PMU counters implemented.

Type: `uint8_t`

Default value: `6`

#### **`subcluster1.ptw_latency`**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.stage12_tlb_size`**

Number of stage1+2 tlb entries. Valid values are 0 or  $\geq 4$ .

Type: `uint32_t`

Default value: `0x80`

#### **`subcluster1.tlb_latency`**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.tlbi\_stall\_enabled**

If true, tlbi invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: `bool`

Default value: `false`

**subcluster1.treat\_PAC\_as\_NOP**

Non-architecture parameter. Treat Pointer Authentication as **NOP**. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are **NOP**. Branch, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

Type: `bool`

Default value: `false`

**subcluster1.walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.CPUCFR**

Value of CPU Configuration Register.

Type: `uint64_t`

Default value: `0x1`

**subcluster2.NUM\_CORES**

Number of cores in the subcluster. Total number of cores in cluster may not exceed 14.

Type: `uint8_t`

Default value: `1`

**subcluster2.cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**subcluster2.cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**subcluster2.cpu0.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster2.cpu0.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster2.cpu0.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster2.cpu0.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu0.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster2.cpu0.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu0.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu0.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu0.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu0.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu0.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

**subcluster2.cpu0.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu0.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu0.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster2.cpu0.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu0.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu0.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100



**subcluster2.cpu0.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster2.cpu0.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu0.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu0.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster2.cpu0.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster2.cpu0.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster2.cpu0.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster2.cpu0.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster2.cpu0.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster2.cpu0.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu0.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu0.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster2.cpu0.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu0.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **subcluster2.cpu0.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **subcluster2.cpu0.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **subcluster2.cpu1.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **subcluster2.cpu1.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

#### **subcluster2.cpu1.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **subcluster2.cpu1.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu1.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter trace\_special\_hlt\_imm16.

Type: bool

Default value: false

**subcluster2.cpu1.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu1.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu1.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu1.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu1.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster2.cpu1.l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: 0x80000

### **subcluster2.cpu1.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **subcluster2.cpu1.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **subcluster2.cpu1.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: `uint8_t`

Default value: 8

### **subcluster2.cpu1.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **subcluster2.cpu1.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster2.cpu1.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster2.cpu1.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster2.cpu1.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu1.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu1.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster2.cpu1.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster2.cpu1.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

### **`subcluster2.cpu1.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **`subcluster2.cpu1.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **`subcluster2.cpu1.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster2.cpu1.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu1.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster2.cpu1.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster2.cpu1.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

#### **`subcluster2.cpu1.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster2.cpu1.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster2.cpu1.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster2.cpu10.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster2.cpu10.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

#### **`subcluster2.cpu10.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`



Default value: `false`

**subcluster2.cpu10.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu10.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster2.cpu10.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu10.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu10.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu10.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **subcluster2.cpu10.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **subcluster2.cpu10.l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: 0x80000

### **subcluster2.cpu10.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **subcluster2.cpu10.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **subcluster2.cpu10.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: `uint8_t`

Default value: 8

### **subcluster2.cpu10.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **subcluster2.cpu10.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster2.cpu10.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

### **subcluster2.cpu10.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

### **subcluster2.cpu10.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

### **subcluster2.cpu10.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

### **subcluster2.cpu10.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster2.cpu10.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster2.cpu10.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster2.cpu10.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster2.cpu10.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster2.cpu10.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster2.cpu10.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu10.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu10.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster2.cpu10.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu10.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster2.cpu10.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster2.cpu10.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster2.cpu11.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster2.cpu11.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster2.cpu11.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster2.cpu11.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu11.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster2.cpu11.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu11.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu11.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **subcluster2.cpu11.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **subcluster2.cpu11.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **subcluster2.cpu11.l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

### **subcluster2.cpu11.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **subcluster2.cpu11.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu11.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: `uint8_t`

Default value: 8

**subcluster2.cpu11.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster2.cpu11.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster2.cpu11.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: 0x100

**subcluster2.cpu11.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: 0

**subcluster2.cpu11.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000



**subcluster2.cpu11.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster2.cpu11.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster2.cpu11.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster2.cpu11.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster2.cpu11.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster2.cpu11.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster2.cpu11.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster2.cpu11.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu11.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu11.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster2.cpu11.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu11.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster2.cpu11.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster2.cpu11.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster2.cpu2.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster2.cpu2.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu2.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu2.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu2.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu2.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu2.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu2.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu2.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu2.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu2.l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

**subcluster2.cpu2.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster2.cpu2.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu2.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster2.cpu2.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu2.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu2.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster2.cpu2.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster2.cpu2.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster2.cpu2.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster2.cpu2.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster2.cpu2.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster2.cpu2.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster2.cpu2.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster2.cpu2.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster2.cpu2.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: true

**subcluster2.cpu2.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu2.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster2.cpu2.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster2.cpu2.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster2.cpu2.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu2.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster2.cpu2.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster2.cpu3.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster2.cpu3.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster2.cpu3.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster2.cpu3.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu3.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.



Type: `bool`

Default value: `false`

### **`subcluster2.cpu3.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu3.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu3.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu3.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu3.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu3.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster2.cpu3.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu3.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu3.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster2.cpu3.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu3.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu3.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster2.cpu3.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster2.cpu3.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu3.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu3.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster2.cpu3.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster2.cpu3.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster2.cpu3.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster2.cpu3.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster2.cpu3.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster2.cpu3.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu3.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu3.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster2.cpu3.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: 0xf000000

**subcluster2.cpu3.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu3.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster2.cpu3.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: true

**subcluster2.cpu4.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: false

**subcluster2.cpu4.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: bool

Default value: false

**subcluster2.cpu4.CRYPTODISABLE**

Disable cryptographic features.

Type: bool

Default value: false

**subcluster2.cpu4.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu4.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster2.cpu4.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu4.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu4.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu4.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu4.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu4.l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

**subcluster2.cpu4.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu4.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu4.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: `uint8_t`

Default value: `8`

**subcluster2.cpu4.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu4.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu4.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster2.cpu4.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster2.cpu4.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu4.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu4.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster2.cpu4.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.



Type: `uint8_t`

Default value: 60

### **`subcluster2.cpu4.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

### **`subcluster2.cpu4.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **`subcluster2.cpu4.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **`subcluster2.cpu4.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster2.cpu4.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu4.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster2.cpu4.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster2.cpu4.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster2.cpu4.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster2.cpu4.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu4.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster2.cpu5.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster2.cpu5.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

#### **`subcluster2.cpu5.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster2.cpu5.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster2.cpu5.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

#### **`subcluster2.cpu5.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster2.cpu5.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster2.cpu5.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu5.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu5.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu5.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster2.cpu5.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu5.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu5.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster2.cpu5.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu5.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu5.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster2.cpu5.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster2.cpu5.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu5.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu5.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster2.cpu5.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster2.cpu5.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster2.cpu5.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster2.cpu5.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster2.cpu5.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster2.cpu5.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu5.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu5.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster2.cpu5.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu5.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster2.cpu5.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster2.cpu5.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster2.cpu6.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster2.cpu6.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster2.cpu6.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster2.cpu6.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu6.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster2.cpu6.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu6.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.



Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu6.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu6.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu6.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu6.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

### **`subcluster2.cpu6.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu6.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu6.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster2.cpu6.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu6.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu6.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster2.cpu6.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster2.cpu6.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster2.cpu6.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster2.cpu6.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster2.cpu6.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster2.cpu6.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster2.cpu6.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster2.cpu6.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster2.cpu6.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster2.cpu6.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu6.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu6.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster2.cpu6.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu6.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster2.cpu6.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu6.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster2.cpu7.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster2.cpu7.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu7.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu7.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu7.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster2.cpu7.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu7.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu7.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu7.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu7.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu7.l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: 0x80000

**subcluster2.cpu7.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu7.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu7.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster2.cpu7.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu7.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu7.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster2.cpu7.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster2.cpu7.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu7.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu7.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster2.cpu7.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster2.cpu7.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster2.cpu7.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string



Default value: N/A

**subcluster2.cpu7.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster2.cpu7.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster2.cpu7.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu7.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu7.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster2.cpu7.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu7.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster2.cpu7.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster2.cpu7.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster2.cpu8.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster2.cpu8.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

#### **`subcluster2.cpu8.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster2.cpu8.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu8.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter trace\_special\_hlt\_imm16.

Type: bool

Default value: false

**subcluster2.cpu8.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu8.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu8.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu8.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu8.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu8.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster2.cpu8.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu8.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu8.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster2.cpu8.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu8.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu8.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster2.cpu8.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster2.cpu8.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu8.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu8.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster2.cpu8.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster2.cpu8.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

### **`subcluster2.cpu8.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **`subcluster2.cpu8.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **`subcluster2.cpu8.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster2.cpu8.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu8.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster2.cpu8.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster2.cpu8.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster2.cpu8.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster2.cpu8.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu8.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster2.cpu9.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster2.cpu9.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu9.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu9.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu9.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu9.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu9.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu9.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu9.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`



Default value: 0x0

### **subcluster2.cpu9.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **subcluster2.cpu9.l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: 0x80000

### **subcluster2.cpu9.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **subcluster2.cpu9.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **subcluster2.cpu9.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: `uint8_t`

Default value: 8

### **subcluster2.cpu9.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **subcluster2.cpu9.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster2.cpu9.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

### **subcluster2.cpu9.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

### **subcluster2.cpu9.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

### **subcluster2.cpu9.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

### **subcluster2.cpu9.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster2.cpu9.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster2.cpu9.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster2.cpu9.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster2.cpu9.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster2.cpu9.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster2.cpu9.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu9.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu9.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster2.cpu9.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu9.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster2.cpu9.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster2.cpu9.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster2.dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.dcache-maintenance\_latency**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.dcache-miss\_latency**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.dcache-prefetch\_enabled**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

Type: bool

Default value: false

**subcluster2.dcache-read\_access\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.dcache-size**

L1 D-Cache size in bytes.

Type: uint32\_t

Default value: 0x10000

#### **subcluster2.dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster2.dcache-write\_access\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster2.dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster2.ecv\_support\_level**

Implement Enhanced Counter Virtualization feature from ARMv8.6. 0, Not supported. 1, fully supported without CNTPOFF. 2, fully supported with CNTPOFF (FEAT\_ECV).

Type: uint8\_t

Default value: 2

#### **subcluster2.ete.CLAIMTAGS**

Number of claim tags.

Type: uint8\_t

Default value: 4

#### **subcluster2.ete.ETE\_REVISION**

ETE revision: 0=ETEv1.0, 1=ETEv1.1, 2=ETEv1.2, 3=ETEv1.3.

Type: uint8\_t

Default value: 1

**subcluster2.ete.MAX\_INST\_PER\_Q**

Maximum limit for the number of instructions implied by a Q element.

Type: `uint16_t`

Default value: 0x1

**subcluster2.ete.NumberOfRSPairs**

Number of resource selector pairs.

Type: `uint8_t`

Default value: 8

**subcluster2.ete.PIDR\_CM0D**

TRCPIDR CM0D value.

Type: `uint8_t`

Default value: 0

**subcluster2.ete.Q\_CADENCE**

Number of instruction blocks traced between two Q elements.

Type: `uint16_t`

Default value: 0x1

**subcluster2.ete.RES0\_STATEFUL**

Whether **RES0** bits are stateful or **RAZ/WI**.

Type: `bool`

Default value: `false`

**subcluster2.ete.RETSTACK**

Return stack depth.

Type: `uint8_t`

Default value: 3

**subcluster2.ete.SIM\_OVERFLOW\_GRANULARITY**

Number of instruction blocks in each granule, for simulated overflow.

Type: `uint32_t`

Default value: 0x64

**subcluster2.ete.SIM\_OVERFLOW\_PERCENTAGE**

Percentage of instruction blocks lost in each granule, for simulated overflow.

Type: uint8\_t

Default value: 0

**subcluster2.ete.SOURCE\_ADDRESS**

Allow generation of source address elements.

Type: bool

Default value: false

**subcluster2.ete.TRACE\_OUTPUT**

File to which to write trace byte stream.

Type: string

Default value: N/A

**subcluster2.ete.TRCRSRTA\_FORCED\_EXCEP**

TRCRSR.TA value for a forcibly traced exception.

Type: bool

Default value: false

**subcluster2.ext\_abort\_so\_write\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: uint8\_t

Default value: 2

**subcluster2.force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks**

Force MTE tag accesses to **RAZ/WI** and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

Type: bool

Default value: false



**subcluster2.force\_zero\_PSTATE\_PAN**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

**subcluster2.force\_zero\_mpam\_partid\_and\_pmg**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

Type: `uint8_t`

Default value: `0`

**subcluster2.has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**subcluster2.has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

**subcluster2.has\_enhanced\_pan**

Implements Armv8.7 Enhanced PAN feature (FEAT\_PAN3) values of this parameter are:- 1, feature is implemented if ARMv8.7 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: `2`

**subcluster2.has\_ete**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (`-plugin` or `-P`).

Type: `bool`

Default value: `false`

**subcluster2.has\_mt\_pmu\_disable\_feature**

Implements Multi-threading PMU disable extension from ARMv8.6 (FEAT\_MTPMU). 0: FEAT\_MTPMU is disabled, 1: FEAT\_MTPMU is enabled if ARMv8.6 is implemented, 2: FEAT\_MTPMU is cherry-picked, 0xF: The feature is disabled and is represented by value 0xF in ID\_AA64DFR0\_EL1.MTPMU.

Type: `uint8_t`

Default value: 0

**subcluster2.has\_statistical\_profiling**

Whether Statistical Based Profiling is implemented (FEAT\_SPE).

Type: `bool`

Default value: `true`

**subcluster2.has\_v8\_7\_spe\_inverted\_filtering**

Where FEAT\_SPEv1p2 is implemented, whether inverted filtering by events is implemented (represented by PMISDR.FnE).

Type: `bool`

Default value: `false`

**subcluster2.has\_v8\_7\_spe\_previous\_branch\_target**

Where FEAT\_SPEv1p2 is implemented, whether the optional branch target feature is implemented (FEAT\_SPE\_PBT).

Type: `bool`

Default value: `false`

**subcluster2.icache-hit\_latency**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster2.icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster2.icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

Type: `bool`

Default value: `false`

**subcluster2.icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when icache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.icache-size**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: `0x10000`

**subcluster2.instruction\_tlb\_size**

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

Type: `uint32_t`

Default value: 0x0

#### **subcluster2.invalidate\_code\_cache\_on\_icache\_cmo**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: uint8\_t

Default value: 0

#### **subcluster2.memory\_tagging\_support\_level**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

Type: uint8\_t

Default value: 3

#### **subcluster2.mpam\_has\_altsp**

DEPRECATED: MPAMIDR\_EL1.HAS\_ALTSP is required when FEAT\_RME is implemented.

Type: bool

Default value: false

#### **subcluster2.mpamidr\_has\_force\_ns**

Whether MPAMIDR\_EL1.HAS\_FORCE\_NS bit is set or clear.values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.6 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

#### **subcluster2.mpamidr\_has\_sdeflt**

Whether MPAMIDR\_EL1.HAS\_SDEFLT bit is set or clear.values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.6 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

#### **subcluster2.mpamidr\_has\_tidr**

Whether MPAMIDR\_EL1.HAS\_TIDR bit is set or clear.values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.6 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

**subcluster2.pmu-num\_counters**

Number of PMU counters implemented.

Type: `uint8_t`

Default value: 31

**subcluster2.ptw\_latency**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

**subcluster2.stage12\_tlb\_size**

Number of stage1+2 tlb entries. Valid values are 0 or >= 4.

Type: `uint32_t`

Default value: 0x80

**subcluster2.tcr\_txsz\_undersize\_should\_fault**

If large VA is not supported, Whether undersized TxSZ value should generate translation fault.

Type: `bool`

Default value: `false`

**subcluster2.tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

**subcluster2.tlbi\_stall\_enabled**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: `bool`

Default value: `false`

**subcluster2.treat\_PAC\_as\_NOP**

Non-architecture parameter. Treat Pointer Authentication as **NOP**. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are **NOP**. Branch, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

Type: `bool`

Default value: `false`

**subcluster2.walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

## 3.35 ARMC1NanoCT\_C1UltraCT

Defined in `LISA/ARMC1NanoCT_C1UltraCT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
C1Nano r0p0	Preliminary support
C1Ultra r0p0	Preliminary support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following parameters were added:

- `num_acp`
- `subcluster0.has_delayed_dbgreg`
- `subcluster0.has_delayed_sysreg`
- `subcluster1.has_delayed_dbgreg`
- `subcluster1.has_delayed_sysreg`
- `subcluster1.mpam_has_altsp`

### About ARMC1NanoCT\_C1UltraCT

The number of cores in each subcluster is configurable using the following parameters:

**subcluster0.NUM\_CORES**

Possible values are 1-13 (ARMC1NanoCT).

**subcluster1.NUM\_CORES**

Possible values are 1-13 (ARMC1UltraCT).

The total number of cores in the cluster cannot exceed 14.

Port arrays are expanded to the width needed for the maximum number of cores in each subcluster. Use the following port array indexes:

- `<port_name>[0-12]` for cores in `subcluster0`.
- `<port_name>[13-25]` for cores in `subcluster1`.



All instances in the Master cross trigger matrix port array, `cti[26]` must be connected, regardless of the `NUM_CORES` value used.

Core-specific parameters have the following prefixes:

- `subcluster0.cpu0` to `subcluster0.cpu12` identify cores in `subcluster0`.
- `subcluster1.cpu0` to `subcluster1.cpu12` identify cores in `subcluster1`.

For information about the cores in this model, see:

- [ARMC1NanoCT](#).
- [ARMC1UltraCT](#).

**Iris and MTI instances for ARMC1NanoCT\_C1UltraCT**

This model has the following Iris instances:

Name	Instance type
ARMC1NanoCT_C1UltraCT	Cluster_ARM_C1-Nano_C1-ULTRA_Heterogeneous
ARMC1NanoCT_C1UltraCT.AMU	PVBusLogger
ARMC1NanoCT_C1UltraCT.AMU.mapper	PVBusMapper
ARMC1NanoCT_C1UltraCT.DAP	PVBusLogger
ARMC1NanoCT_C1UltraCT.DAP.mapper	PVBusMapper
ARMC1NanoCT_C1UltraCT.DSU	C1-DSU
ARMC1NanoCT_C1UltraCT.DSU.PPU_cluster	PPUv1
ARMC1NanoCT_C1UltraCT.DSU.PPU_cluster.busslave	PVBusSlave
ARMC1NanoCT_C1UltraCT.DSU.PPU_coreZ (where Z = 0-1)	PPUv1
ARMC1NanoCT_C1UltraCT.DSU.PPU_coreZ.busslave (where Z = 0-1)	PVBusSlave
ARMC1NanoCT_C1UltraCT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMC1NanoCT_C1UltraCT.DSU.mpam_busslave	PVBusSlave
ARMC1NanoCT_C1UltraCT.DSU.shared_cache	PVCache
ARMC1NanoCT_C1UltraCT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster

Name	Instance type
ARMC1NanoCT_C1UltraCT.DSU.shared_cache.upstream[Z] (where Z = 0-6)	PVBusSlave
ARMC1NanoCT_C1UltraCT.DSU.utility_slave[0]	PVBusSlave
ARMC1NanoCT_C1UltraCT.MMAP	PVBusLogger
ARMC1NanoCT_C1UltraCT.MMAP.mapper	PVBusMapper
ARMC1NanoCT_C1UltraCT.RAS	PVBusLogger
ARMC1NanoCT_C1UltraCT.RAS.mapper	PVBusMapper
ARMC1NanoCT_C1UltraCT.cpuZ.debug_rom (where Z = 0-1)	debug_rom
ARMC1NanoCT_C1UltraCT.ext_bus	PVBusLogger
ARMC1NanoCT_C1UltraCT.ext_bus.mapper	PVBusMapper
ARMC1NanoCT_C1UltraCT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMC1NanoCT_C1UltraCT.global_debug_rom	debug_rom
ARMC1NanoCT_C1UltraCT.secondary_debug_rom	debug_rom
ARMC1NanoCT_C1UltraCT.subcluster0	Subcluster_ARM_C1-Nano
ARMC1NanoCT_C1UltraCT.subcluster0.cpu0	ARM_C1-Nano
ARMC1NanoCT_C1UltraCT.subclusterZ.cpu0.UTLB (where Z = 0-1)	TLB
ARMC1NanoCT_C1UltraCT.subclusterZ.cpuU.dtlb (where Z = 0-1; U = 0-1)	TLB
ARMC1NanoCT_C1UltraCT.subclusterZ.cpu0.l1dcache (where Z = 0-1)	PVCache
ARMC1NanoCT_C1UltraCT.subclusterZ.cpu0.l1dcache.downstream[0].pvbusmaster (where Z = 0-1)	PVBusMaster
ARMC1NanoCT_C1UltraCT.subclusterZ.cpu0.l1dcache.upstream[0] (where Z = 0-1)	PVBusSlave
ARMC1NanoCT_C1UltraCT.subclusterZ.cpu0.l1icache (where Z = 0-1)	PVCache
ARMC1NanoCT_C1UltraCT.subclusterZ.cpu0.l1icache.downstream[0].pvbusmaster (where Z = 0-1)	PVBusMaster
ARMC1NanoCT_C1UltraCT.subclusterZ.cpu0.l1icache.upstream[0] (where Z = 0-1)	PVBusSlave
ARMC1NanoCT_C1UltraCT.subclusterZ.cpu0.l2cache (where Z = 0-1)	PVCache
ARMC1NanoCT_C1UltraCT.subclusterZ.cpu0.l2cache.downstream[0].pvbusmaster (where Z = 0-1)	PVBusMaster
ARMC1NanoCT_C1UltraCT.subclusterZ.cpu0.l2cache.upstream[W] (where Z = 0-1; W = 0-1)	PVBusSlave
ARMC1NanoCT_C1UltraCT.subclusterZ.sve (where Z = 0-1)	ScalableVectorExtension
ARMC1NanoCT_C1UltraCT.subcluster1	Subcluster_ARM_C1-Ultra
ARMC1NanoCT_C1UltraCT.subcluster1.cpu0	ARM_C1-Ultra

This model has the following MTI trace components:

Name	Component type
ARMC1NanoCT_C1UltraCT.AMU	PVBusLogger
ARMC1NanoCT_C1UltraCT.AMU.mapper	PVBusMapper
ARMC1NanoCT_C1UltraCT.DAP	PVBusLogger
ARMC1NanoCT_C1UltraCT.DAP.mapper	PVBusMapper
ARMC1NanoCT_C1UltraCT.DSU	C1-DSU



Name	Component type
ARMC1NanoCT_C1UltraCT.DSU.PPU_cluster	PPUV1
ARMC1NanoCT_C1UltraCT.DSU.PPU_cluster.busslave	PVBusSlave
ARMC1NanoCT_C1UltraCT.DSU.PPU_coreZ (where Z = 0-1)	PPUV1
ARMC1NanoCT_C1UltraCT.DSU.PPU_coreZ.busslave (where Z = 0-1)	PVBusSlave
ARMC1NanoCT_C1UltraCT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMC1NanoCT_C1UltraCT.DSU.mpam_busslave	PVBusSlave
ARMC1NanoCT_C1UltraCT.DSU.shared_cache	PVCache
ARMC1NanoCT_C1UltraCT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMC1NanoCT_C1UltraCT.DSU.shared_cache.upstream[Z] (where Z = 0-6)	PVBusSlave
ARMC1NanoCT_C1UltraCT.DSU.utility_slave[0]	PVBusSlave
ARMC1NanoCT_C1UltraCT.MMAP	PVBusLogger
ARMC1NanoCT_C1UltraCT.MMAP.mapper	PVBusMapper
ARMC1NanoCT_C1UltraCT.RAS	PVBusLogger
ARMC1NanoCT_C1UltraCT.RAS.mapper	PVBusMapper
ARMC1NanoCT_C1UltraCT.ext_bus	PVBusLogger
ARMC1NanoCT_C1UltraCT.ext_bus.mapper	PVBusMapper
ARMC1NanoCT_C1UltraCT.gic_cpuif_decoder_cluster	GCv3CPUInterfaceDecoder
ARMC1NanoCT_C1UltraCT.subcluster0.cpu0	ARM_C1-Nano
ARMC1NanoCT_C1UltraCT.subclusterZ.cpu0.UTLB (where Z = 0-1)	TLB
ARMC1NanoCT_C1UltraCT.subclusterZ.cpu0.l1dcache (where Z = 0-1)	PVCache
ARMC1NanoCT_C1UltraCT.subclusterZ.cpu0.l1dcache.downstream[0].pvbusmaster (where Z = 0-1)	PVBusMaster
ARMC1NanoCT_C1UltraCT.subclusterZ.cpu0.l1dcache.upstream[0] (where Z = 0-1)	PVBusSlave
ARMC1NanoCT_C1UltraCT.subclusterZ.cpu0.l1icache (where Z = 0-1)	PVCache
ARMC1NanoCT_C1UltraCT.subclusterZ.cpu0.l1icache.downstream[0].pvbusmaster (where Z = 0-1)	PVBusMaster
ARMC1NanoCT_C1UltraCT.subclusterZ.cpu0.l1icache.upstream[0] (where Z = 0-1)	PVBusSlave
ARMC1NanoCT_C1UltraCT.subclusterZ.cpu0.l2cache (where Z = 0-1)	PVCache
ARMC1NanoCT_C1UltraCT.subclusterZ.cpu0.l2cache.downstream[0].pvbusmaster (where Z = 0-1)	PVBusMaster
ARMC1NanoCT_C1UltraCT.subclusterZ.cpu0.l2cache.upstream[W] (where Z = 0-1; W = 0-1)	PVBusSlave
ARMC1NanoCT_C1UltraCT.subcluster1.cpu0	ARM_C1-Ultra

## Ports for ARMC1NanoCT\_C1UltraCT

Port	Direction	Protocol	Description
acp_s	slave	PVBus	AXI ACP subordinate port.
AEND0MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND2MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).

Port	Direction	Protocol	Description
AEND3MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART2MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	slave	Signal	CHI defined pins.
broadcastcachemaint	slave	Signal	ACE defined pins.
broadcastouter	slave	Signal	ACE defined pins.
broadcastpersist	slave	Signal	CHI defined pins.
cfgend	slave	Signal	This signal is for EE bit initialisation.
cfgsdisable	slave	Signal	This signal disables write access to some secure Interrupt Controller registers.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcs_m_pchannel	master	PChannel	Cluster PCSM signal
cluster_powerdown_out	master	Signal	DEPRECATED - An output from PPUs that informs thermal controller of the cluster power info
cluster_ppu_hw_stat	master	Value	Cluster PPU power state
clusterid	slave	Value	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set ManagerID64 of the core.
clusterpmuirq	master	Signal	DynamiQ pmu irq
cme_pcs_m_pchannel	master	PChannel	Cluster PCSM signal
cmeerrirq	master	Signal	RAS cme err irq
cmefaultirq	master	Signal	RAS cme fault irq
CNTHPIRQ	master	Signal	Timer signals to SOC.
CNTHPSIRQ	master	Signal	Timer signals to SOC
CNTHVIRQ	master	Signal	Timer signals to SOC.
CNTHVSIRQ	master	Signal	Timer signals to SOC
CNTPNSIRQ	master	Signal	Timer signals to SOC.
CNTPSIRQ	master	Signal	Timer signals to SOC.

Port	Direction	Protocol	Description
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.
CNTVIRQ	master	Signal	Timer signals to SOC.
commirq	master	Signal	Interrupt signal from debug communications channel.
core_clk_in	slave	ClockSignal	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
core_pcsn_pchannel	master	PChannel	Core PCSM signals
core_powerdown_out	master	Signal	DEPRECATED - An output from PPUs that informs thermal controller of the core power info
core_ppu_hw_stat	master	Value	Notify the power state change inside the PPU.
coreerrirq	master	Signal	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq	master	Signal	Fault indicator for a detected 1 or 2 bit ECC error
coreinstrrun	master	Signal	Core Running State
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
ctidbgirq	master	Signal	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	slave	Signal	External debug interface.
dbgnopwrdown	master	Signal	No power-down request.
dbgpwrupreq	master	Signal	Debug power up request.
defaultMP	slave	Signal	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main manager interface ports.
dev_debug_s	slave	PVBus	External debug interface.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.
gicreset	master	Signal	An output from PPUs that can be used to reset GICCLK domain components
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	slave	PVBus	External debug interface.
pmbirq	master	Signal	Interrupt signal from the statistical profiling unit.
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	master	Signal	PPU cluster interrupt.
ppu_cluster_isolate	master	Signal	PPU Cluster Domain isolation control.
ppu_cluster_wakerequest	slave	Signal	PPU cluster wake request signal.
ppu_cme_irq	master	Signal	PPU CME interrupt
ppu_core_irq	master	Signal	PPU core interrupt.
ppu_core_isolate	master	Signal	PPU Core Domain isolation control.
ppu_core_wakerequest	slave	Signal	PPU core wake request signal.

Port	Direction	Protocol	Description
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	master	PVBus	The core will generate bus requests on this port.
pvbus_periph_m	master	PVBus	The core can generate peripheral bus request on this port.
rei	slave	Signal	Per core RAM Error Interrupt.
reset	slave	Signal	Reset signal to cluster.
rvbaraddr	slave	Value_64	Reset vector base address.
sci_m	master	SystemCoherencyInterface	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei	slave	Signal	Per core System Error physical pins.
spiden	slave	Signal	External debug interface.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq	master	Signal	Interrupt signal from the trace buffer unit.
utility_bus_s	slave	PVBus	Utility bus subordinate
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfiq	slave	Signal	Virtualised FIQ.
virq	slave	Signal	Virtualised IRQ.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Per core virtual System Error physical pins.

## Parameters for ARMC1NanoCT\_C1UltraCT

### AEND0\_DEFAULT

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

Type: uint64\_t

Default value: 0x0

### AEND1\_DEFAULT

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

Type: uint64\_t

Default value: 0x0

**AEND2\_DEFAULT**

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

Type: `uint64_t`

Default value: `0x0`

**AEND3\_DEFAULT**

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART0\_DEFAULT**

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART1\_DEFAULT**

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART2\_DEFAULT**

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART3\_DEFAULT**

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

Type: `uint64_t`

Default value: `0x0`

**BROADCASTATOMIC**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`

Default value: `true`

**BROADCASTCACHEMAINT**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

Type: `bool`

Default value: `false`

**BROADCASTOUTER**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

Type: `bool`

Default value: `false`

**BROADCASTPERSIST**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: `bool`

Default value: `true`

**CLUSTER\_ID**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the ManagerID64 of the core.

Type: `uint32_t`

Default value: `0x0`

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. This value may affect performance of the simulation, but

is required if the rest of the system is sensitive to certain even when cache state modelling is disabled. Possible values are:- 0 = All CMOs are broadcast if architecturally required- 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required.

Type: `uint8_t`

Default value: 1

#### **DBGROMADDR**

Initialization value of DBGDRAR register. Bits[55:12] of this register specify the ROM table physical address.

Type: `uint64_t`

Default value: 0x0

#### **DBGROMADDRV**

If true, set bits[1:0] of the CP15 DBGDRAR to indicate that the address is valid.

Type: `bool`

Default value: `false`

#### **GICDISABLE**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`

Default value: `true`

#### **NUM\_CMES**

Defines how many CMEs are associated with each cluster. This parameter must be in sync with SVE.ScalableVectorExtension SME configuration.

Type: `uint8_t`

Default value: 0

#### **bus\_type**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

Type: `uint8_t`

Default value: 0

**core\_complex\_mapping**

Defines Complex descriptions for platforms that support several Cores per Complex like Cortex-A510. JSON format: {"complex0": { "cores" : [ 0, 1 ], "l2-cache" :{"exists":1, "size":16MB}}, ... , "complexN": { "cores" : [<core\_list>], "l2-cache" : {"exists":1, "size":16MB}}} where <core\_list> is the list of cores in the complexN. Effective only when the parameter value is not empty.

Type: string

Default value: "{ \"complex0\": { \"cores\": [0, 1], \"l2-cache\" :{ \"exists\":1, \"size\": \"16MB\"}}, \"complex1\": { \"cores\": [2, 3], \"l2-cache\" :{ \"exists\":1, \"size\": \"16MB\"}}, \"complex2\": { \"cores\": [4, 5], \"l2-cache\" :{ \"exists\":1, \"size\": \"16MB\"}}, \"complex3\": { \"cores\": [6, 7], \"l2-cache\" :{ \"exists\":1, \"size\": \"16MB\"}}, \"complex4\": { \"cores\": [8, 9], \"l2-cache\" :{ \"exists\":1, \"size\": \"16MB\"}}, \"complex5\": { \"cores\": [10, 11], \"l2-cache\" :{ \"exists\":1, \"size\": \"16MB\"}}, \"complex6\": { \"cores\": [12, 13], \"l2-cache\" :{ \"exists\":1, \"size\": \"16MB\"}} }"

**core\_power\_on\_by\_default**

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPUs which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

Type: bool

Default value: false

**dcache-state\_modelled**

Set whether D-cache has stateful implementation.

Type: bool

Default value: false

**enable\_simulation\_performance\_optimizations**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

Type: bool

Default value: true

**has\_peripheral\_port**

If true, an additional AXI peripheral port is configured.

Type: bool



Default value: `false`

### **`icache-state_modelled`**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **`l3cache-has_mpam`**

L3 Cache has MPAM support.

Type: `bool`

Default value: `true`

### **`l3cache-hit_latency`**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-maintenance_latency`**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-miss_latency`**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-mpamf.max_partid_ns`**

Maximum value of non-secure PARTID supported for DSU L3Cache. Options are 7 or 63.

Type: `uint8_t`

Default value: 63

**l3cache-read\_access\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size l3cache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**l3cache-read\_bus\_width\_in\_bytes**

L3 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**l3cache-read\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. l3cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**l3cache-size**

L3 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**l3cache-snoop\_data\_transfer\_latency**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**l3cache-snoop\_issue\_latency**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **l3cache-ways**

L3 Cache number of ways (sets are implicit from size).

Type: `uint16_t`

Default value: 16

### **l3cache-write\_access\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size `l3cache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **l3cache-write\_bus\_width\_in\_bytes**

L3 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: `uint32_t`

Default value: 0x10

### **l3cache-write\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l3cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **mpmm\_accumulator\_multiplier**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator. value of `n` means the accumulator will use  $(n * \text{accumulator value})$  to calculate the mpmm threshold (MPMM). is provided as a fast model workaround to handle cases where execution of quantums in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

Type: `uint8_t`

Default value: 1

### **num\_acp**

Number of ACP ports.

Type: `uint8_t`

Default value: 0

### **num\_nodes**

Number of transport nodes. Zero implies direct-connect configuration.

Type: `uint8_t`

Default value: 1

### **scu\_cache\_protection**

SCU-L3 is configured with ECC if true.

Type: `bool`

Default value: `true`

### **subcluster0.CPUCFR**

Value of CPU Configuration Register.

Type: `uint64_t`

Default value: `0x0`

### **subcluster0.NUM\_CORES**

Number of cores in the subcluster. Total number of cores in cluster may not exceed 14.

Type: `uint8_t`

Default value: 1

### **subcluster0.core\_cache\_protection**

`core_cache_protection` can change `ERROFR`, `ERROPFGF` and `ERROPFGCTL` fields. Possible values are: -1:Not implemented (by default), 0:Disabled, 1:Enabled.

Type: `int8_t`

Default value: 1

### **subcluster0.cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**subcluster0.cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**subcluster0.cpu0.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster0.cpu0.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster0.cpu0.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu0.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu0.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu0.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu0.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu0.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu0.l2cache-read_bus_width_in_bytes`**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: `uint32_t`

Default value: `0x10`

### **`subcluster0.cpu0.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu0.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

**subcluster0.cpu0.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu0.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu0.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu0.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu0.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu0.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu0.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu0.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster0.cpu0.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t



Default value: 171

**subcluster0.cpu0.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu0.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu0.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu0.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu0.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu0.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu0.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: 0xf000000

**subcluster0.cpu0.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu0.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster0.cpu0.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: true

**subcluster0.cpu1.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: false

**subcluster0.cpu1.CRYPTODISABLE**

Disable cryptographic features.

Type: bool

Default value: false

**subcluster0.cpu1.RVBARADDR**

Value of RVBAR\_ELx register.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter trace\_special\_hlt\_imm16.

Type: bool

Default value: false

**subcluster0.cpu1.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu1.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu1.l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

**subcluster0.cpu1.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu1.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu1.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: `uint8_t`

Default value: `8`

**subcluster0.cpu1.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu1.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu1.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu1.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu1.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu1.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu1.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**`subcluster0.cpu1.semihosting-T32_HLT`**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**`subcluster0.cpu1.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**`subcluster0.cpu1.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**`subcluster0.cpu1.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**`subcluster0.cpu1.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**`subcluster0.cpu1.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**`subcluster0.cpu1.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf0000000`

### **`subcluster0.cpu1.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster0.cpu1.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf0000000`

### **`subcluster0.cpu1.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu1.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu1.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu10.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster0.cpu10.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster0.cpu10.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu10.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu10.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu10.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu10.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`



**subcluster0.cpu10.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu10.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu10.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu10.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu10.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu10.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu10.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

#### **subcluster0.cpu10.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu10.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

#### **subcluster0.cpu10.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu10.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

#### **subcluster0.cpu10.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu10.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu10.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu10.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu10.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster0.cpu10.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster0.cpu10.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu10.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu10.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: true

**subcluster0.cpu10.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu10.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu10.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster0.cpu10.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu10.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu10.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu10.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu11.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu11.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu11.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu11.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu11.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu11.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu11.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu11.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu11.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: `uint32_t`

Default value: `0x10`

**subcluster0.cpu11.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu11.l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: 0x80000

**subcluster0.cpu11.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu11.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu11.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu11.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu11.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu11.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu11.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu11.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu11.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu11.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu11.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu11.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster0.cpu11.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.



Type: `uint8_t`

Default value: 171

### **`subcluster0.cpu11.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **`subcluster0.cpu11.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **`subcluster0.cpu11.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster0.cpu11.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu11.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu11.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster0.cpu11.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

#### **`subcluster0.cpu11.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster0.cpu11.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu11.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster0.cpu12.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu12.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu12.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: 0x0

#### **subcluster0.cpu12.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter trace\_special\_hlt\_imm16.

Type: bool

Default value: false

#### **subcluster0.cpu12.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu12.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu12.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu12.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu12.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

#### **subcluster0.cpu12.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu12.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

#### **subcluster0.cpu12.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu12.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu12.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

#### **subcluster0.cpu12.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu12.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu12.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu12.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu12.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu12.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu12.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu12.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu12.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster0.cpu12.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster0.cpu12.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu12.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu12.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu12.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu12.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu12.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu12.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu12.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu12.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster0.cpu12.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster0.cpu2.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster0.cpu2.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster0.cpu2.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu2.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu2.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu2.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`



**subcluster0.cpu2.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu2.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu2.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu2.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu2.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu2.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu2.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu2.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu2.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu2.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster0.cpu2.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster0.cpu2.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster0.cpu2.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster0.cpu2.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu2.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu2.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu2.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu2.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu2.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu2.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu2.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu3.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu3.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu3.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu3.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu3.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu3.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu3.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu3.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu3.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu3.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu3.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu3.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu3.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu3.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.



Type: `uint8_t`

Default value: 60

### **`subcluster0.cpu3.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

### **`subcluster0.cpu3.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **`subcluster0.cpu3.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **`subcluster0.cpu3.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster0.cpu3.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu3.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu3.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster0.cpu3.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu3.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu3.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu3.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu4.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu4.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster0.cpu4.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu4.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu4.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu4.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu4.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu4.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster0.cpu4.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu4.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu4.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu4.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu4.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu4.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu4.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu4.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu4.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu4.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster0.cpu4.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster0.cpu4.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu4.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu4.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu4.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu4.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster0.cpu4.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu4.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu4.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster0.cpu4.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu5.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu5.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu5.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu5.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu5.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu5.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`



**subcluster0.cpu5.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu5.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu5.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu5.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu5.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu5.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu5.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu5.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu5.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu5.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster0.cpu5.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster0.cpu5.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster0.cpu5.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster0.cpu5.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu5.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu5.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu5.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu5.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu5.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster0.cpu5.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu5.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster0.cpu6.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu6.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu6.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu6.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu6.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu6.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu6.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu6.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu6.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu6.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu6.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu6.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu6.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu6.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.



Type: `uint8_t`

Default value: 60

### **`subcluster0.cpu6.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

### **`subcluster0.cpu6.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **`subcluster0.cpu6.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **`subcluster0.cpu6.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster0.cpu6.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: 0x0

### **`subcluster0.cpu6.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: 0xf000000

### **`subcluster0.cpu6.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster0.cpu6.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu6.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu6.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu6.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu7.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu7.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu7.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu7.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu7.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu7.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu7.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu7.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster0.cpu7.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu7.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu7.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu7.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu7.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu7.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu7.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu7.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu7.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu7.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster0.cpu7.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster0.cpu7.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu7.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu7.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu7.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu7.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu7.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu7.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu7.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu7.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster0.cpu7.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu8.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu8.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu8.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu8.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu8.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu8.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`



**subcluster0.cpu8.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu8.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu8.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu8.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu8.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu8.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu8.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu8.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu8.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu8.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster0.cpu8.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster0.cpu8.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster0.cpu8.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster0.cpu8.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu8.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu8.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu8.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu8.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu8.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster0.cpu8.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu8.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster0.cpu9.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu9.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu9.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu9.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu9.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu9.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu9.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu9.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu9.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu9.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu9.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu9.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu9.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu9.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.



Type: `uint8_t`

Default value: 60

**`subcluster0.cpu9.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**`subcluster0.cpu9.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**`subcluster0.cpu9.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**`subcluster0.cpu9.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**`subcluster0.cpu9.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**`subcluster0.cpu9.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**`subcluster0.cpu9.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster0.cpu9.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu9.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu9.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu9.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.dcache-hit_latency`**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.dcache-maintenance_latency`**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

#### **subcluster0.dcache-miss\_latency**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.dcache-prefetch\_enabled**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

Type: bool

Default value: false

#### **subcluster0.dcache-read\_access\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.dcache-size**

L1 D-Cache size in bytes.

Type: uint32\_t

Default value: 0x8000

#### **subcluster0.dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.dcache-write_access_latency`**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.dcache-write_latency`**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.ete.CLAIMTAGS`**

Number of claim tags.

Type: `uint8_t`

Default value: 4

#### **`subcluster0.ete.MAX_INST_PER_Q`**

Maximum limit for the number of instructions implied by a Q element.

Type: `uint16_t`

Default value: `0x1`

#### **`subcluster0.ete.PIDR_CM0D`**

TRCPIDR CM0D value.

Type: `uint8_t`

Default value: 0

#### **`subcluster0.ete.PIDR_REVAND`**

TRCPIDR REVAND value.

Type: `uint8_t`

Default value: 0

**subcluster0.ete.PIDR\_REVISION**

TRCPIDR REVISION value.

Type: `uint8_t`

Default value: 0

**subcluster0.ete.Q\_CADENCE**

Number of instruction blocks traced between two Q elements.

Type: `uint16_t`

Default value: 0x1

**subcluster0.ete.RES0\_STATEFUL**

Whether **RES0** bits are stateful or **RAZ/WI**.

Type: `bool`

Default value: `false`

**subcluster0.ete.RETSTACK**

Return stack depth.

Type: `uint8_t`

Default value: 3

**subcluster0.ete.REVISION**

TRCIDR1 revision value.

Type: `uint8_t`

Default value: 0

**subcluster0.ete.SIM\_OVERFLOW\_GRANULARITY**

Number of instruction blocks in each granule, for simulated overflow.

Type: `uint32_t`

Default value: 0x64

**subcluster0.ete.SIM\_OVERFLOW\_PERCENTAGE**

Percentage of instruction blocks lost in each granule, for simulated overflow.

Type: `uint8_t`

Default value: 0

**subcluster0.ete.SOURCE\_ADDRESS**

Allow generation of source address elements.

Type: bool

Default value: false

**subcluster0.ete.TRACE\_OUTPUT**

File to which to write trace byte stream.

Type: string

Default value: N/A

**subcluster0.ete.TRCRSRTA\_FORCED\_EXCEP**

TRCRSR.TA value for a forcibly traced exception.

Type: bool

Default value: false

**subcluster0.force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks**

Force MTE tag accesses to **RAZ/WI** and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

Type: bool

Default value: false

**subcluster0.force\_zero\_PSTATE\_PAN**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: bool

Default value: false

**subcluster0.force\_zero\_mpam\_partid\_and\_pmg**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

Type: uint8\_t

Default value: 0

**subcluster0.has\_actlr2**

If true ACLTR2 exists and ACTLR2(NS) is aliased to ACTLR\_EL1[63:32].

Type: bool

Default value: true

**subcluster0.has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: bool

Default value: false

**subcluster0.has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: bool

Default value: true

**subcluster0.has\_dot\_product**

Possible values of this parameter are:- 1, feature not implemented.- 2, feature is implemented.

Type: uint8\_t

Default value: 2

**subcluster0.has\_ete**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (-plugin or -P).

Type: bool

Default value: false

**subcluster0.icache-hit\_latency**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

Type: bool

Default value: false

**subcluster0.icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.icache-size**

L1 I-Cache size in bytes.



Type: `uint32_t`

Default value: `0x8000`

#### **`subcluster0.invalidate_code_cache_on_icache_cmo`**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: `uint8_t`

Default value: `0`

#### **`subcluster0.memory_tagging_support_level`**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

Type: `uint8_t`

Default value: `3`

#### **`subcluster0.pmu-num_counters`**

Number of PMU counters implemented.

Type: `uint8_t`

Default value: `6`

#### **`subcluster0.ptw_latency`**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.ras_pfg_clock_mhz`**

RAS Pseudo-Fault generation clock rate in MHz.

Type: `uint8_t`

Default value: `12`

#### **`subcluster0.tlb_latency`**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.tlbi\_stall\_enabled**

If true, tlbi invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: `bool`

Default value: `false`

**subcluster0.treat\_PAC\_as\_NOP**

Non-architecture parameter. Treat Pointer Authentication as **NOP**. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are **NOP**. Branch, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

Type: `bool`

Default value: `false`

**subcluster0.walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.CPUCFR**

Value of CPU Configuration Register.

Type: `uint64_t`

Default value: `0x1`

**subcluster1.NUM\_CORES**

Number of cores in the subcluster. Total number of cores in cluster may not exceed 14.

Type: `uint8_t`

Default value: `1`

**subcluster1.cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**subcluster1.cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**subcluster1.cpu0.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu0.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster1.cpu0.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu0.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu0.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu0.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu0.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu0.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu0.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu0.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu0.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

**subcluster1.cpu0.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu0.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu0.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster1.cpu0.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu0.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu0.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu0.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu0.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu0.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu0.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu0.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster1.cpu0.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster1.cpu0.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster1.cpu0.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu0.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu0.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu0.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu0.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu0.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu0.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster1.cpu0.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu0.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster1.cpu1.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu1.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu1.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu1.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`



**subcluster1.cpu1.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter trace\_special\_hlt\_imm16.

Type: bool

Default value: false

**subcluster1.cpu1.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu1.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu1.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu1.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu1.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu1.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster1.cpu1.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu1.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu1.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster1.cpu1.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu1.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu1.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu1.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu1.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu1.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu1.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu1.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster1.cpu1.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

### **`subcluster1.cpu1.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **`subcluster1.cpu1.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **`subcluster1.cpu1.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster1.cpu1.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster1.cpu1.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster1.cpu1.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

#### **`subcluster1.cpu1.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster1.cpu1.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu1.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster1.cpu10.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu10.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu10.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu10.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu10.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu10.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu10.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu10.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu10.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

#### **subcluster1.cpu10.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu10.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

#### **subcluster1.cpu10.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu10.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu10.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

#### **subcluster1.cpu10.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster1.cpu10.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster1.cpu10.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

### **subcluster1.cpu10.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

### **subcluster1.cpu10.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

### **subcluster1.cpu10.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

### **subcluster1.cpu10.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456



**subcluster1.cpu10.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster1.cpu10.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster1.cpu10.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster1.cpu10.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster1.cpu10.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu10.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu10.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu10.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu10.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu10.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu10.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu10.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster1.cpu11.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu11.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster1.cpu11.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu11.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu11.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu11.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu11.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu11.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu11.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu11.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu11.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

#### **`subcluster1.cpu11.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu11.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu11.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: `uint8_t`

Default value: 8

**subcluster1.cpu11.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster1.cpu11.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster1.cpu11.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: 0x100

**subcluster1.cpu11.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: 0

**subcluster1.cpu11.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster1.cpu11.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu11.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu11.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster1.cpu11.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster1.cpu11.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu11.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu11.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu11.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu11.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu11.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster1.cpu11.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu11.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu11.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster1.cpu11.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster1.cpu12.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu12.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu12.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu12.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu12.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu12.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`



**subcluster1.cpu12.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu12.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu12.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu12.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu12.l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

**subcluster1.cpu12.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster1.cpu12.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu12.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster1.cpu12.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu12.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu12.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu12.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu12.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster1.cpu12.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster1.cpu12.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: 0x123456

**subcluster1.cpu12.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster1.cpu12.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster1.cpu12.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster1.cpu12.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster1.cpu12.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: true

**subcluster1.cpu12.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu12.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu12.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster1.cpu12.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu12.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu12.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu12.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster1.cpu2.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu2.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster1.cpu2.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu2.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu2.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu2.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu2.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu2.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu2.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster1.cpu2.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu2.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu2.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster1.cpu2.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu2.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu2.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu2.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu2.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu2.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu2.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu2.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster1.cpu2.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t



Default value: 171

**subcluster1.cpu2.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu2.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu2.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu2.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu2.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu2.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: 0xf000000

**subcluster1.cpu2.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu2.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster1.cpu2.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: true

**subcluster1.cpu3.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: false

**subcluster1.cpu3.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: bool

Default value: false

**subcluster1.cpu3.CRYPTODISABLE**

Disable cryptographic features.

Type: bool

Default value: false

**subcluster1.cpu3.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu3.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster1.cpu3.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster1.cpu3.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu3.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu3.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu3.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu3.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu3.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

### **`subcluster1.cpu3.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

### **`subcluster1.cpu3.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **`subcluster1.cpu3.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **`subcluster1.cpu3.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster1.cpu3.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu3.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster1.cpu3.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster1.cpu3.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster1.cpu3.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu3.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu3.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster1.cpu4.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster1.cpu4.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu4.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu4.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu4.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu4.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu4.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu4.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`



**subcluster1.cpu4.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu4.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu4.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster1.cpu4.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu4.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu4.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster1.cpu4.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu4.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu4.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu4.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu4.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu4.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu4.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu4.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster1.cpu4.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster1.cpu4.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu4.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu4.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu4.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu4.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu4.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu4.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu4.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu4.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu4.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster1.cpu5.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu5.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster1.cpu5.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu5.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu5.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu5.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu5.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu5.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu5.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu5.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu5.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

### **`subcluster1.cpu5.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu5.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu5.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster1.cpu5.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu5.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu5.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu5.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu5.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu5.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu5.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu5.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster1.cpu5.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster1.cpu5.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu5.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`



**subcluster1.cpu5.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu5.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu5.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu5.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu5.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu5.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu5.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu5.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster1.cpu6.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster1.cpu6.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu6.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu6.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu6.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu6.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu6.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu6.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu6.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu6.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu6.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster1.cpu6.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu6.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu6.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster1.cpu6.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu6.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu6.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu6.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu6.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu6.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu6.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu6.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster1.cpu6.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster1.cpu6.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster1.cpu6.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster1.cpu6.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu6.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu6.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu6.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu6.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu6.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster1.cpu6.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu6.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster1.cpu7.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu7.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu7.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu7.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu7.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter trace\_special\_hlt\_imm16.

Type: bool

Default value: false

**subcluster1.cpu7.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu7.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu7.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu7.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu7.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t



Default value: 0x0

**subcluster1.cpu7.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster1.cpu7.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu7.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu7.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster1.cpu7.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu7.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu7.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu7.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu7.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu7.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu7.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu7.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster1.cpu7.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

### **`subcluster1.cpu7.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **`subcluster1.cpu7.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **`subcluster1.cpu7.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster1.cpu7.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu7.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster1.cpu7.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster1.cpu7.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

#### **`subcluster1.cpu7.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster1.cpu7.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu7.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster1.cpu8.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu8.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu8.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu8.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu8.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu8.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu8.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu8.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu8.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

#### **subcluster1.cpu8.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu8.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

#### **subcluster1.cpu8.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu8.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu8.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

#### **subcluster1.cpu8.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster1.cpu8.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster1.cpu8.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

### **subcluster1.cpu8.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

### **subcluster1.cpu8.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

### **subcluster1.cpu8.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

### **subcluster1.cpu8.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu8.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster1.cpu8.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster1.cpu8.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster1.cpu8.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster1.cpu8.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu8.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: 0x0

**subcluster1.cpu8.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: 0xf000000



**subcluster1.cpu8.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu8.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu8.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu8.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu8.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster1.cpu9.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu9.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster1.cpu9.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu9.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu9.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu9.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu9.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu9.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu9.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu9.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu9.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

#### **`subcluster1.cpu9.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu9.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu9.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster1.cpu9.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu9.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu9.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu9.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu9.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu9.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster1.cpu9.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster1.cpu9.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu9.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu9.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu9.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu9.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster1.cpu9.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu9.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu9.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster1.cpu9.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster1.dcache-hit_latency`**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.dcache-maintenance_latency`**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.dcache-miss_latency`**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.dcache-prefetch_enabled`**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

Type: `bool`

Default value: `false`

### **`subcluster1.dcache-read_access_latency`**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.dcache-size**

L1 D-Cache size in bytes.

Type: `uint32_t`

Default value: `0x10000`

**subcluster1.dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.dcache-write\_access\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.ecv\_support\_level**

Implement Enhanced Counter Virtualization feature from ARMv8.6. 0, Not supported. 1, fully supported without CNTPOFF. 2, fully supported with CNTPOFF (FEAT\_ECV).



Type: `uint8_t`

Default value: 2

#### **`subcluster1.ete.CLAIMTAGS`**

Number of claim tags.

Type: `uint8_t`

Default value: 4

#### **`subcluster1.ete.ETE_REVISION`**

ETE revision: 0=ETEv1.0, 1=ETEv1.1, 2=ETEv1.2, 3=ETEv1.3.

Type: `uint8_t`

Default value: 1

#### **`subcluster1.ete.MAX_INST_PER_Q`**

Maximum limit for the number of instructions implied by a Q element.

Type: `uint16_t`

Default value: 0x1

#### **`subcluster1.ete.NumberOfRSPairs`**

Number of resource selector pairs.

Type: `uint8_t`

Default value: 8

#### **`subcluster1.ete.PIDR_CM0D`**

TRCPIDR CM0D value.

Type: `uint8_t`

Default value: 0

#### **`subcluster1.ete.Q_CADENCE`**

Number of instruction blocks traced between two Q elements.

Type: `uint16_t`

Default value: 0x1

#### **`subcluster1.ete.RES0_STATEFUL`**

Whether **RES0** bits are stateful or **RAZ/WI**.

Type: `bool`

Default value: `false`

#### **`subcluster1.ete.RETSTACK`**

Return stack depth.

Type: `uint8_t`

Default value: 3

#### **`subcluster1.ete.SIM_OVERFLOW_GRANULARITY`**

Number of instruction blocks in each granule, for simulated overflow.

Type: `uint32_t`

Default value: 0x64

#### **`subcluster1.ete.SIM_OVERFLOW_PERCENTAGE`**

Percentage of instruction blocks lost in each granule, for simulated overflow.

Type: `uint8_t`

Default value: 0

#### **`subcluster1.ete.SOURCE_ADDRESS`**

Allow generation of source address elements.

Type: `bool`

Default value: `false`

#### **`subcluster1.ete.TRACE_OUTPUT`**

File to which to write trace byte stream.

Type: `string`

Default value: N/A

#### **`subcluster1.ete.TRCRSRTA_FORCED_EXCEP`**

TRCRSR.TA value for a forcibly traced exception.

Type: `bool`

Default value: `false`

**subcluster1.ext\_abort\_so\_write\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO , 4 = UER, 5 = CE.

Type: `uint8_t`

Default value: 2

**subcluster1.force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks**

Force MTE tag accesses to **RAZ/WI** and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

Type: `bool`

Default value: `false`

**subcluster1.force\_zero\_PSTATE\_PAN**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

**subcluster1.force\_zero\_mpam\_partid\_and\_pmg**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAMO\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

Type: `uint8_t`

Default value: 0

**subcluster1.has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**subcluster1.has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

**subcluster1.has\_enhanced\_pan**

Implements Armv8.7 Enhanced PAN feature (FEAT\_PAN3) values of this parameter are:- 1, feature is implemented if ARMv8.7 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: `2`

**subcluster1.has\_ete**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (-plugin or -P).

Type: `bool`

Default value: `false`

**subcluster1.has\_mt\_pmu\_disable\_feature**

Implements Multi-threading PMU disable extension from ARMv8.6 (FEAT\_MTPMU). 0: FEAT\_MTPMU is disabled, 1: FEAT\_MTPMU is enabled if ARMv8.6 is implemented, 2: FEAT\_MTPMU is cherry-picked, 0xF: The feature is disabled and is represented by value 0xF in ID\_AA64DFR0\_EL1.MTPMU.

Type: `uint8_t`

Default value: `0`

**subcluster1.has\_statistical\_profiling**

Whether Statistical Based Profiling is implemented (FEAT\_SPE).

Type: `bool`

Default value: `true`

**subcluster1.has\_v8\_7\_spe\_inverted\_filtering**

Where FEAT\_SPEv1p2 is implemented, whether inverted filtering by events is implemented (represented by PMISDR.FnE).

Type: `bool`

Default value: `false`

**subcluster1.has\_v8\_7\_spe\_previous\_branch\_target**

Where FEAT\_SPEv1p2 is implemented, whether the optional branch target feature is implemented (FEAT\_SPE\_PBT).

Type: `bool`

Default value: `false`

**subcluster1.icache-hit\_latency**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

**subcluster1.icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to

the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.icache-size`**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: `0x10000`

#### **`subcluster1.instruction_tlb_size`**

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

Type: `uint32_t`

Default value: `0x0`

#### **`subcluster1.invalidate_code_cache_on_icache_cmo`**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: `uint8_t`

Default value: `0`

#### **`subcluster1.memory_tagging_support_level`**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

Type: `uint8_t`

Default value: `3`

#### **`subcluster1.mpam_has_altsp`**

DEPRECATED: MPAMIDR\_EL1.HAS\_ALTSP is required when FEAT\_RME is implemented.

Type: `bool`

Default value: `false`

**subcluster1.mpamidr\_has\_force\_ns**

Whether MPAMIDR\_EL1.HAS\_FORCE\_NS bit is set or clear.values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.6 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

**subcluster1.mpamidr\_has\_sdeflt**

Whether MPAMIDR\_EL1.HAS\_SDEFLT bit is set or clear.values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.6 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

**subcluster1.mpamidr\_has\_tidr**

Whether MPAMIDR\_EL1.HAS\_TIDR bit is set or clear.values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.6 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

**subcluster1.pmu-num\_counters**

Number of PMU counters implemented.

Type: uint8\_t

Default value: 31

**subcluster1.ptw\_latency**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

**subcluster1.stage12\_tlb\_size**

Number of stage1+2 tlb entries. Valid values are 0 or >= 4.

Type: uint32\_t

Default value: 0x80

**subcluster1.tcr\_txsz\_undersize\_should\_fault**

If large VA is not supported, Whether undersized TxSZ value should generate translation fault.

Type: `bool`

Default value: `false`

**subcluster1.tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.tlbi\_stall\_enabled**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: `bool`

Default value: `false`

**subcluster1.treat\_PAC\_as\_NOP**

Non-architecture parameter. Treat Pointer Authentication as **NOP**. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are **NOP**. Branch, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

Type: `bool`

Default value: `false`

**subcluster1.walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

## 3.36 ARMC1PremiumCT

Defined in `LISA/ARMC1PremiumCT.lisa`.

This model supports the following revisions of the IP at the given quality levels:



Revision	Quality level
r0p0	Preliminary support

For an explanation of the quality levels, see [Quality level definitions](#).

## Changes in 11.31.15

The following parameters were added:

- `has_delayed_dbgreg`
- `has_delayed_sysreg`
- `mpam_has_altsp`
- `num_acp`

## About ARMC1PremiumCT

The following example platforms are available:

- FVP\_Base\_C1-Premium
- EVS\_Dhrystone\_C1-Premiumx1
- SVP\_Base\_C1-Premiumx1

The following functionality is supported in this release:

- C1-SME2 is supported with the following limitations:
  - The arbitration and assignment logic is not supported.
  - RAS error handling is supported in the first CME only.
  - It is assumed that C1-SME2 units are operating in dynamic mode, with a minimum power mode of OFF. Accompanying PPU registers may be read or written to, but they do not determine the behavior of the CME. The PPU interrupt signal for the C1-SME2 is also not modeled or exposed at this time.
- Pilatus DSU support for C1-SME2.
- L2Cache is supported at the per-core level only.
- BROADCASTPERSIST pin is implemented.
- Optional peripheral port is supported.
- L3Cache partition is supported.
- Per-core clock is supported.

## Limitations

The following features are not yet supported, and will be added in a future release:

- BROADCASTCACHEMAINTPOU pin is not implemented.
- COREINSTRRET signals are not implemented.
- MEM\_RET power mode is not supported.

The following features are not implemented:

- DynamIQ features that are negligible to the programmers' view simulation will not be implemented in the Fast Model.
- 256-bit wide output transactions will not be supported.
- Error correction/detection features will not be supported.
- Self-test features (MBIST) will not be supported.
- Latency configuration will not be supported.
- Snoop filtering will not be supported.
- Cache stashing capability will not be supported.

### Iris and MTI instances for ARMC1PremiumCT

This model has the following Iris instances:

Name	Instance type
ARMC1PremiumCT	Cluster_ARM_C1-Premium
ARMC1PremiumCT.AMU	PVBusLogger
ARMC1PremiumCT.AMU.mapper	PVBusMapper
ARMC1PremiumCT.DAP	PVBusLogger
ARMC1PremiumCT.DAP.mapper	PVBusMapper
ARMC1PremiumCT.DSU	C1-DSU
ARMC1PremiumCT.DSU.PPU_cluster	PPUv1
ARMC1PremiumCT.DSU.PPU_cluster.busslave	PVBusSlave
ARMC1PremiumCT.DSU.PPU_core0	PPUv1
ARMC1PremiumCT.DSU.PPU_core0.busslave	PVBusSlave
ARMC1PremiumCT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMC1PremiumCT.DSU.mpam_busslave	PVBusSlave
ARMC1PremiumCT.DSU.shared_cache	PVCache
ARMC1PremiumCT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMC1PremiumCT.DSU.shared_cache.upstream[Y] (where Y = 0-4)	PVBusSlave
ARMC1PremiumCT.DSU.utility_slave[0]	PVBusSlave
ARMC1PremiumCT.MMAP	PVBusLogger
ARMC1PremiumCT.MMAP.mapper	PVBusMapper
ARMC1PremiumCT.RAS	PVBusLogger
ARMC1PremiumCT.RAS.mapper	PVBusMapper
ARMC1PremiumCT.cpu0	ARM_C1-Premium
ARMC1PremiumCT.cpu0.UTLB	TLB
ARMC1PremiumCT.cpu0.debug_rom	debug_rom
ARMC1PremiumCT.cpu0.dtlb	TLB
ARMC1PremiumCT.cpu0.l1dcache	PVCache
ARMC1PremiumCT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMC1PremiumCT.cpu0.l1dcache.upstream[0]	PVBusSlave

Name	Instance type
ARMC1PremiumCT.cpu0.l1icache	PVCache
ARMC1PremiumCT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMC1PremiumCT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMC1PremiumCT.cpu0.l2cache	PVCache
ARMC1PremiumCT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMC1PremiumCT.cpu0.l2cache.upstream[U] (where U = 0-1)	PVBusSlave
ARMC1PremiumCT.ext_bus	PVBusLogger
ARMC1PremiumCT.ext_bus.mapper	PVBusMapper
ARMC1PremiumCT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMC1PremiumCT.global_debug_rom	debug_rom
ARMC1PremiumCT.secondary_debug_rom	debug_rom
ARMC1PremiumCT.sve	ScalableVectorExtension

This model has the following MTI trace components:

Name	Component type
ARMC1PremiumCT.AMU	PVBusLogger
ARMC1PremiumCT.AMU.mapper	PVBusMapper
ARMC1PremiumCT.DAP	PVBusLogger
ARMC1PremiumCT.DAP.mapper	PVBusMapper
ARMC1PremiumCT.DSU	C1-DSU
ARMC1PremiumCT.DSU.PPU_cluster	PPUv1
ARMC1PremiumCT.DSU.PPU_cluster.busslave	PVBusSlave
ARMC1PremiumCT.DSU.PPU_core0	PPUv1
ARMC1PremiumCT.DSU.PPU_core0.busslave	PVBusSlave
ARMC1PremiumCT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMC1PremiumCT.DSU.mpam_busslave	PVBusSlave
ARMC1PremiumCT.DSU.shared_cache	PVCache
ARMC1PremiumCT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMC1PremiumCT.DSU.shared_cache.upstream[Y] (where Y = 0-4)	PVBusSlave
ARMC1PremiumCT.DSU.utility_slave[0]	PVBusSlave
ARMC1PremiumCT.MMAP	PVBusLogger
ARMC1PremiumCT.MMAP.mapper	PVBusMapper
ARMC1PremiumCT.RAS	PVBusLogger
ARMC1PremiumCT.RAS.mapper	PVBusMapper
ARMC1PremiumCT.cpu0	ARM_C1-Premium
ARMC1PremiumCT.cpu0.UTLB	TLB
ARMC1PremiumCT.cpu0.l1dcache	PVCache
ARMC1PremiumCT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMC1PremiumCT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMC1PremiumCT.cpu0.l1icache	PVCache

Name	Component type
ARMC1PremiumCT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMC1PremiumCT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMC1PremiumCT.cpu0.l2cache	PVCache
ARMC1PremiumCT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMC1PremiumCT.cpu0.l2cache.upstream[U] (where U = 0-1)	PVBusSlave
ARMC1PremiumCT.ext_bus	PVBusLogger
ARMC1PremiumCT.ext_bus.mapper	PVBusMapper
ARMC1PremiumCT.gic_cpuif_decoder_cluster	GLCv3CPUInterfaceDecoder

## Ports for ARMC1PremiumCT

Port	Direction	Protocol	Description
acp_s	slave	PVBus	AXI ACP subordinate port.
AEND0MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND2MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND3MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART2MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	slave	Signal	CHI defined pins.
broadcastcachemaint	slave	Signal	ACE defined pins.
broadcastouter	slave	Signal	ACE defined pins.
broadcastpersist	slave	Signal	CHI defined pins.
cfgend	slave	Signal	This signal is for EE bit initialisation.
cfgsdisable	slave	Signal	This signal disables write access to some secure Interrupt Controller registers.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcsn_pchannel	master	PChannel	Cluster PCSM signal
cluster_powerdown_out	master	Signal	DEPRECATED - An output from PPUs that informs thermal controller of the cluster power info
cluster_ppu_hw_stat	master	Value	Cluster PPU power state

Port	Direction	Protocol	Description
clustercriticalirq	master	Signal	Cluster Critical Irq
clustererrirq	master	Signal	Cluster Error Irq
clusterfaultirq	master	Signal	Cluster Fault Irq
clusterid	slave	Value	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set ManagerID64 of the core.
clusterpmuirq	master	Signal	DynamiQ pmu irq
cmeerrirq	master	Signal	RAS cme err irq
cmefaultirq	master	Signal	RAS cme fault irq
CNTHPIRQ	master	Signal	Timer signals to SOC.
CNTHPSIRQ	master	Signal	Timer signals to SOC
CNTHVIRQ	master	Signal	Timer signals to SOC.
CNTHVSIQ	master	Signal	Timer signals to SOC
CNTPNSIRQ	master	Signal	Timer signals to SOC.
CNTPSIQ	master	Signal	Timer signals to SOC.
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.
CNTVIRQ	master	Signal	Timer signals to SOC.
commirq	master	Signal	Interrupt signal from debug communications channel.
core_clk_in	slave	ClockSignal	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
core_pcsn_pchannel	master	PChannel	Core PCSM signals
core_powerdown_out	master	Signal	DEPRECATED - An output from PPUs that informs thermal controller of the core power info
core_ppu_hw_stat	master	Value	Notify the power state change inside the PPU.
coreerrirq	master	Signal	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq	master	Signal	Fault indicator for a detected 1 or 2 bit ECC error
coreinstrrun	master	Signal	Core Running State
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
ctidbgirq	master	Signal	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	slave	Signal	External debug interface.
dbgnopwrdown	master	Signal	No power-down request.
dbgpwrupreq	master	Signal	Debug power up request.

Port	Direction	Protocol	Description
defaultMP	slave	Signal	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main manager interface ports.
dev_debug_s	slave	PVBus	External debug interface.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.
gicreset	master	Signal	An output from PPU that can be used to reset GICCLK domain components
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	slave	PVBus	External debug interface.
pmbirq	master	Signal	Interrupt signal from the statistical profiling unit.
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	master	Signal	PPU cluster interrupt.
ppu_cluster_isolate	master	Signal	PPU Cluster Domain isolation control.
ppu_cluster_wakerequest	slave	Signal	PPU cluster wake request signal.
ppu_core_irq	master	Signal	PPU core interrupt.
ppu_core_isolate	master	Signal	PPU Core Domain isolation control.
ppu_core_wakerequest	slave	Signal	PPU core wake request signal.
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	master	PVBus	The core will generate bus requests on this port.
pvbus_periph_m	master	PVBus	The core can generate peripheral bus request on this port.
rei	slave	Signal	Per core RAM Error Interrupt.
reset	slave	Signal	Reset signal to cluster.
rvbaraddr	slave	Value_64	Reset vector base address.
sci_m	master	SystemCoherencyInterface	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei	slave	Signal	Per core System Error physical pins.
spiden	slave	Signal	External debug interface.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq	master	Signal	Interrupt signal from the trace buffer unit.
utility_bus_s	slave	PVBus	Utility bus subordinate
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfiq	slave	Signal	Virtualised FIQ.
virq	slave	Signal	Virtualised IRQ.

Port	Direction	Protocol	Description
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Per core virtual System Error physical pins.

## Parameters for ARMC1PremiumCT

### **cpuX.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **cpuX.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **cpuX.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **cpuX.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **cpuX.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **cpuX.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x100000`



**cpuX.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**cpuX.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**cpuX.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**cpuX.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**cpuX.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**cpuX.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**cpuX.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**cpuX.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**cpuX.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**cpuX.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**cpuX.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**cpuX.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**cpuX.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf0000000`

**cpuX.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**cpuX.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf0000000`

**cpuX.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`cpuX.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`cpuX.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`AEND0_DEFAULT`**

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

Type: `uint64_t`

Default value: `0x0`

### **`AEND1_DEFAULT`**

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

Type: `uint64_t`

Default value: `0x0`

### **`AEND2_DEFAULT`**

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

Type: `uint64_t`

Default value: `0x0`

### **`AEND3_DEFAULT`**

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

Type: `uint64_t`

Default value: 0x0

#### **ASTART0\_DEFAULT**

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

Type: uint64\_t

Default value: 0x0

#### **ASTART1\_DEFAULT**

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

Type: uint64\_t

Default value: 0x0

#### **ASTART2\_DEFAULT**

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

Type: uint64\_t

Default value: 0x0

#### **ASTART3\_DEFAULT**

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

Type: uint64\_t

Default value: 0x0

#### **BROADCASTATOMIC**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: bool

Default value: true

#### **BROADCASTCACHEMAINT**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

Type: bool

Default value: `false`

### **BROADCASTOUTER**

Enable broadcasting of Outer Shareable transactions. The `broadcastouter` signal will override this value if used.

Type: `bool`

Default value: `false`

### **BROADCASTPERSIST**

Enable broadcasting of cache clean to the point of persistence operations. The `broadcastpersist` signal will override this value if used.

Type: `bool`

Default value: `true`

### **CLUSTER\_ID**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the ManagerID64 of the core.

Type: `uint32_t`

Default value: `0x0`

### **CMO\_broadcast\_when\_cache\_state\_modelling\_disabled**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. This value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required.

Type: `uint8_t`

Default value: `1`

### **CPUCFR**

Value of CPU Configuration Register.

Type: `uint64_t`

Default value: `0x1`

**DSU.cme.CMECFR**

Value of CMECFR\_EL1 fields (ECC, CHI) for all CMEs.

Type: `string`

Default value: `"{\\"ECC\\": 0, \\"CHI\\": 0}"`

**DSU.cme.mpmm\_accumulator\_multiplier**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator. value of `n` means the accumulator will use  $(n * \text{accumulator value})$  to calculate the mpmm threshold (MPMM). is provided as a fast model workaround to handle cases where execution of quantum in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

Type: `uint8_t`

Default value: `1`

**DSU.cme.power\_on\_by\_default**

If true, CME PPU's will initialize in Dynamic mode out of reset, enabling the CME to power itself on/off automatically as it is used.

Type: `bool`

Default value: `true`

**GICDISABLE**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`

Default value: `true`

**NUM\_CMES**

Defines how many CMEs are associated with each cluster. This parameter must be in sync with SVE.ScalableVectorExtension SME configuration.

Type: `uint8_t`

Default value: `0`

**NUM\_CORES**

Number of cores per cluster.

Type: `uint8_t`

Default value: `1`

**bus\_type**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

Type: `uint8_t`

Default value: 0

**core\_power\_on\_by\_default**

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPUs which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

Type: `bool`

Default value: `false`

**cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-maintenance\_latency**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`



**dcache-miss\_latency**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-prefetch\_enabled**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

Type: `bool`

Default value: `false`

**dcache-read\_access\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-state\_modelled**

Set whether D-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **`dcache-write_access_latency`**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`dcache-write_latency`**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`default_opmode`**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

Type: `uint8_t`

Default value: 4

### **`diagnostics`**

Enable DynamIQ diagnostic messages.

Type: `bool`

Default value: `false`

### **`ecv_support_level`**

Implement Enhanced Counter Virtualization feature from ARMv8.6. 0, Not supported. 1, fully supported without CNTPOFF. 2, fully supported with CNTPOFF (FEAT\_ECV).

Type: `uint8_t`

Default value: 2

### **`enable_simulation_performance_optimizations`**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences

seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

Type: `bool`

Default value: `true`

### **ete.CLAIMTAGS**

Number of claim tags.

Type: `uint8_t`

Default value: 4

### **ete.ETE\_REVISION**

ETE revision: 0=ETEv1.0, 1=ETEv1.1, 2=ETEv1.2, 3=ETEv1.3.

Type: `uint8_t`

Default value: 1

### **ete.MAX\_INST\_PER\_Q**

Maximum limit for the number of instructions implied by a Q element.

Type: `uint16_t`

Default value: 0x1

### **ete.NumberOfRSPairs**

Number of resource selector pairs.

Type: `uint8_t`

Default value: 8

### **ete.PIDR\_CM0D**

TRCPIDR CM0D value.

Type: `uint8_t`

Default value: 0

### **ete.Q\_CADENCE**

Number of instruction blocks traced between two Q elements.

Type: `uint16_t`

Default value: 0x1

**ete.RES0\_STATEFUL**

Whether **RES0** bits are stateful or **RAZ/WI**.

Type: `bool`

Default value: `false`

**ete.RETSTACK**

Return stack depth.

Type: `uint8_t`

Default value: `3`

**ete.SIM\_OVERFLOW\_GRANULARITY**

Number of instruction blocks in each granule, for simulated overflow.

Type: `uint32_t`

Default value: `0x64`

**ete.SIM\_OVERFLOW\_PERCENTAGE**

Percentage of instruction blocks lost in each granule, for simulated overflow.

Type: `uint8_t`

Default value: `0`

**ete.SOURCE\_ADDRESS**

Allow generation of source address elements.

Type: `bool`

Default value: `false`

**ete.TRACE\_OUTPUT**

File to which to write trace byte stream.

Type: `string`

Default value: `N/A`

**ete.TRCSRSTA\_FORCED\_EXCEP**

TRCSR.TA value for a forcibly traced exception.

Type: `bool`

Default value: `false`

**ext\_abort\_so\_write\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO , 4 = UER, 5 = CE.

Type: `uint8_t`

Default value: 2

**force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks**

Force MTE tag accesses to **RAZ/WI** and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

Type: `bool`

Default value: `false`

**force\_zero\_PSTATE\_PAN**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

**force\_zero\_mpam\_partid\_and\_pmg**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAMO\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

Type: `uint8_t`

Default value: 0

**has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

**has\_enhanced\_pan**

Implements Armv8.7 Enhanced PAN feature (FEAT\_PAN3) values of this parameter are:- 1, feature is implemented if ARMv8.7 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 2

**has\_ete**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (-plugin or -P).

Type: `bool`

Default value: `false`

**has\_mt\_pmu\_disable\_feature**

Implements Multi-threading PMU disable extension from ARMv8.6 (FEAT\_MTPMU). 0: FEAT\_MTPMU is disabled, 1: FEAT\_MTPMU is enabled if ARMv8.6 is implemented, 2: FEAT\_MTPMU is cherry-picked, 0xF: The feature is disabled and is represented by value 0xF in ID\_AA64DFR0\_EL1.MTPMU.

Type: `uint8_t`

Default value: 0

**has\_peripheral\_port**

If true, an additional AXI peripheral port is configured.

Type: `bool`

Default value: `false`

**has\_statistical\_profiling**

Whether Statistical Based Profiling is implemented (FEAT\_SPE).

Type: `bool`

Default value: `true`

**has\_v8\_7\_spe\_inverted\_filtering**

Where FEAT\_SPEv1p2 is implemented, whether inverted filtering by events is implemented (represented by PMISDR.FnE).

Type: `bool`

Default value: `false`

**has\_v8\_7\_spe\_previous\_branch\_target**

Where FEAT\_SPEv1p2 is implemented, whether the optional branch target feature is implemented (FEAT\_SPE\_PBT).

Type: `bool`

Default value: `false`

**icache-hit\_latency**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

**icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **icache-state\_modelled**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **instruction\_tlb\_size**

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

Type: `uint32_t`

Default value: 0x0

### **invalidate\_code\_cache\_on\_icache\_cmo**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: `uint8_t`

Default value: 0

### **l3cache-mpamf.max\_partid\_ns**

Maximum value of non-secure PARTID supported for DSU L3Cache. Options are 7 or 63.

Type: `uint8_t`

Default value: 63

### **l3cache-size**

L3 Cache size in bytes.

Type: `uint32_t`

Default value: 0x100000



**log2\_trace\_buffer\_alignment**

Log2 of trace buffer alignment constraint for output buffer (0->1B ... 11->2Kib).

Type: `uint8_t`

Default value: 6

**memory\_tagging\_support\_level**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

Type: `uint8_t`

Default value: 3

**mpam\_has\_altsp**

DEPRECATED: MPAMIDR\_EL1.HAS\_ALTSP is required when FEAT\_RME is implemented.

Type: `bool`

Default value: `false`

**mpamidr\_has\_force\_ns**

Whether MPAMIDR\_EL1.HAS\_FORCE\_NS bit is set or clear.values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.6 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

**mpamidr\_has\_sdeflt**

Whether MPAMIDR\_EL1.HAS\_SDEFLT bit is set or clear.values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.6 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

**mpamidr\_has\_tidr**

Whether MPAMIDR\_EL1.HAS\_TIDR bit is set or clearvalues of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.6 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

**mpmm\_accumulator\_multiplier**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator.value of n means the accumulator will use (n \* accumulator value) to calculate the mpmm threshold (MPMM).is provided as a fast model workaround to handle cases where execution of quantum in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

Type: uint8\_t

Default value: 1

**num\_acp**

Number of ACP ports.

Type: uint8\_t

Default value: 0

**num\_nodes**

Number of transport nodes. Zero implies direct-connect configuration.

Type: uint8\_t

Default value: 1

**pmu-num\_counters**

Number of PMU counters implemented.

Type: uint8\_t

Default value: 31

**ptw\_latency**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

**stage12\_tlb\_size**

Number of stage1+2 tlb entries. Valid values are 0 or >= 4.

Type: uint32\_t

Default value: 0x80

**tcr\_txsz\_undersize\_should\_fault**

If large VA is not supported, Whether undersized TxSZ value should generate translation fault.

Type: `bool`

Default value: `false`

**tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

**tlbi\_stall\_enabled**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: `bool`

Default value: `false`

**treat\_PAC\_as\_NOP**

Non-architecture parameter. Treat Pointer Authentication as **NOP**. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are **NOP**. Branch, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

Type: `bool`

Default value: `false`

**walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

## 3.37 ARMC1ProCT

Defined in `LISA/ARMC1ProCT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Preliminary support

For an explanation of the quality levels, see [Quality level definitions](#).

## Changes in 11.31.15

The following parameters were added:

- `has_delayed_dbgreg`
- `has_delayed_sysreg`
- `num_acp`

## About ARMC1ProCT

The following example platforms are available:

- FVP\_Base\_C1-Pro
- EVS\_Dhrystone\_C1-Prox1
- SVP\_Base\_C1-Prox1

The following functionality is supported in this release:

- C1-SME2 is supported with the following limitations:
  - The arbitration and assignment logic is not supported.
  - RAS error handling is supported in the first CME only.
  - The C1-SME2 auxiliary AMU counters are not supported.
  - It is assumed that C1-SME2 units are operating in dynamic mode, with a minimum power mode of OFF. Accompanying PPU registers may be read or written to, but they do not determine the behavior of the CME. The PPU interrupt signal for the C1-SME2 is also not modeled or exposed at this time.
- Pilatus DSU support for C1-SME2.
- L2Cache is supported at the per-core level only.
- BROADCASTPERSIST pin is implemented.
- Optional peripheral port is supported.
- L3Cache partition is supported.
- Per-core clock is supported.

## Limitations

The following features are not yet supported, and will be added in a future release:

- BROADCASTCACHEMAINTPOU pin is not implemented.
- COREINSTRRET signals are not implemented.
- MEM\_RET power mode is not supported.

The following features are not implemented:

- DynamIQ features that are negligible to the programmers' view simulation will not be implemented in the Fast Model.
- 256-bit wide output transactions will not be supported.
- Error correction/detection features will not be supported.
- Self-test features (MBIST) will not be supported.
- Latency configuration will not be supported.
- Snoop filtering will not be supported.
- Cache stashing capability will not be supported.

### Iris and MTI instances for ARMC1ProCT

This model has the following Iris instances:

Name	Instance type
ARMC1ProCT	Cluster_ARM_C1-Pro
ARMC1ProCT.AMU	PVBusLogger
ARMC1ProCT.AMU.mapper	PVBusMapper
ARMC1ProCT.DAP	PVBusLogger
ARMC1ProCT.DAP.mapper	PVBusMapper
ARMC1ProCT.DSU	C1-DSU
ARMC1ProCT.DSU.PPU_cluster	PPUv1
ARMC1ProCT.DSU.PPU_cluster.busslave	PVBusSlave
ARMC1ProCT.DSU.PPU_core0	PPUv1
ARMC1ProCT.DSU.PPU_core0.busslave	PVBusSlave
ARMC1ProCT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMC1ProCT.DSU.mpam_busslave	PVBusSlave
ARMC1ProCT.DSU.shared_cache	PVCache
ARMC1ProCT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMC1ProCT.DSU.shared_cache.upstream[Y] (where Y = 0-4)	PVBusSlave
ARMC1ProCT.DSU.utility_slave[0]	PVBusSlave
ARMC1ProCT.MMAP	PVBusLogger
ARMC1ProCT.MMAP.mapper	PVBusMapper
ARMC1ProCT.RAS	PVBusLogger
ARMC1ProCT.RAS.mapper	PVBusMapper
ARMC1ProCT.cpu0	ARM_C1-Pro
ARMC1ProCT.cpu0.UTLB	TLB
ARMC1ProCT.cpu0.debug_rom	debug_rom
ARMC1ProCT.cpu0.dtlb	TLB
ARMC1ProCT.cpu0.l1dcache	PVCache
ARMC1ProCT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMC1ProCT.cpu0.l1dcache.upstream[0]	PVBusSlave

Name	Instance type
ARMC1ProCT.cpu0.l1icache	PVCache
ARMC1ProCT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMC1ProCT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMC1ProCT.cpu0.l2cache	PVCache
ARMC1ProCT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMC1ProCT.cpu0.l2cache.upstream[U] (where $U = 0-1$ )	PVBusSlave
ARMC1ProCT.ext_bus	PVBusLogger
ARMC1ProCT.ext_bus.mapper	PVBusMapper
ARMC1ProCT.gic_cpuif_decoder_cluster	GCv3CPUInterfaceDecoder
ARMC1ProCT.global_debug_rom	debug_rom
ARMC1ProCT.secondary_debug_rom	debug_rom
ARMC1ProCT.sve	ScalableVectorExtension

This model has the following MTI trace components:

Name	Component type
ARMC1ProCT.AMU	PVBusLogger
ARMC1ProCT.AMU.mapper	PVBusMapper
ARMC1ProCT.DAP	PVBusLogger
ARMC1ProCT.DAP.mapper	PVBusMapper
ARMC1ProCT.DSU	C1-DSU
ARMC1ProCT.DSU.PPU_cluster	PPUv1
ARMC1ProCT.DSU.PPU_cluster.busslave	PVBusSlave
ARMC1ProCT.DSU.PPU_core0	PPUv1
ARMC1ProCT.DSU.PPU_core0.busslave	PVBusSlave
ARMC1ProCT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMC1ProCT.DSU.mpam_busslave	PVBusSlave
ARMC1ProCT.DSU.shared_cache	PVCache
ARMC1ProCT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMC1ProCT.DSU.shared_cache.upstream[Y] (where $Y = 0-4$ )	PVBusSlave
ARMC1ProCT.DSU.utility_slave[0]	PVBusSlave
ARMC1ProCT.MMAP	PVBusLogger
ARMC1ProCT.MMAP.mapper	PVBusMapper
ARMC1ProCT.RAS	PVBusLogger
ARMC1ProCT.RAS.mapper	PVBusMapper
ARMC1ProCT.cpu0	ARM_C1-Pro
ARMC1ProCT.cpu0.UTLB	TLB
ARMC1ProCT.cpu0.l1dcache	PVCache
ARMC1ProCT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMC1ProCT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMC1ProCT.cpu0.l1icache	PVCache

Name	Component type
ARMC1ProCT.cpu0.l1licache.downstream[0].pvbusmaster	PVBusMaster
ARMC1ProCT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMC1ProCT.cpu0.l2cache	PVCache
ARMC1ProCT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMC1ProCT.cpu0.l2cache.upstream[U] (where U = 0-1)	PVBusSlave
ARMC1ProCT.ext_bus	PVBusLogger
ARMC1ProCT.ext_bus.mapper	PVBusMapper
ARMC1ProCT.gic_cpuif_decoder_cluster	GLCv3CPUInterfaceDecoder

## Ports for ARMC1ProCT

Port	Direction	Protocol	Description
acp_s	slave	PVBus	AXI ACP subordinate port.
AEND0MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND2MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND3MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART2MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	slave	Signal	CHI defined pins.
broadcastcachemaint	slave	Signal	ACE defined pins.
broadcastouter	slave	Signal	ACE defined pins.
broadcastpersist	slave	Signal	CHI defined pins.
cfgend	slave	Signal	This signal is for EE bit initialisation.
cfgsdisable	slave	Signal	This signal disables write access to some secure Interrupt Controller registers.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcsn_pchannel	master	PChannel	Cluster PCSM signal
cluster_powerdown_out	master	Signal	DEPRECATED - An output from PPUs that informs thermal controller of the cluster power info
cluster_ppu_hw_stat	master	Value	Cluster PPU power state

Port	Direction	Protocol	Description
clusterid	slave	Value	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set ManagerID64 of the core.
clusterpmuirq	master	Signal	DynamlQ pmu irq
cme_pcs_m_pchannel	master	PChannel	Cluster PCSM signal
cmeerrirq	master	Signal	RAS cme err irq
cmefaultirq	master	Signal	RAS cme fault irq
CNTHPIRQ	master	Signal	Timer signals to SOC.
CNTHPSIRQ	master	Signal	Timer signals to SOC
CNTHVIRQ	master	Signal	Timer signals to SOC.
CNTHVSIRQ	master	Signal	Timer signals to SOC
CNTPNSIRQ	master	Signal	Timer signals to SOC.
CNTPSIRQ	master	Signal	Timer signals to SOC.
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.
CNTVIRQ	master	Signal	Timer signals to SOC.
commirq	master	Signal	Interrupt signal from debug communications channel.
core_clk_in	slave	ClockSignal	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
core_pcs_m_pchannel	master	PChannel	Core PCSM signals
core_powerdown_out	master	Signal	DEPRECATED - An output from PPUs that informs thermal controller of the core power info
core_ppu_hw_stat	master	Value	Notify the power state change inside the PPU.
coreerrirq	master	Signal	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq	master	Signal	Fault indicator for a detected 1 or 2 bit ECC error
coreinstrrun	master	Signal	Core Running State
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
ctidbgirq	master	Signal	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	slave	Signal	External debug interface.
dbgnopwrdown	master	Signal	No power-down request.
dbgpwrupreq	master	Signal	Debug power up request.
defaultMP	slave	Signal	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main manager interface ports.
dev_debug_s	slave	PVBus	External debug interface.



Port	Direction	Protocol	Description
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.
gicreset	master	Signal	An output from PPU that can be used to reset GICCLK domain components
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	slave	PVBus	External debug interface.
pmbirq	master	Signal	Interrupt signal from the statistical profiling unit.
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	master	Signal	PPU cluster interrupt.
ppu_cluster_isolate	master	Signal	PPU Cluster Domain isolation control.
ppu_cluster_wakerequest	slave	Signal	PPU cluster wake request signal.
ppu_cme_irq	master	Signal	PPU CME interrupt
ppu_core_irq	master	Signal	PPU core interrupt.
ppu_core_isolate	master	Signal	PPU Core Domain isolation control.
ppu_core_wakerequest	slave	Signal	PPU core wake request signal.
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	master	PVBus	The core will generate bus requests on this port.
pvbus_periph_m	master	PVBus	The core can generate peripheral bus request on this port.
rei	slave	Signal	Per core RAM Error Interrupt.
reset	slave	Signal	Reset signal to cluster.
rvbaraddr	slave	Value_64	Reset vector base address.
sci_m	master	SystemCoherencyInterface	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei	slave	Signal	Per core System Error physical pins.
spiden	slave	Signal	External debug interface.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq	master	Signal	Interrupt signal from the trace buffer unit.
utility_bus_s	slave	PVBus	Utility bus subordinate
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfiq	slave	Signal	Virtualised FIQ.
virq	slave	Signal	Virtualised IRQ.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Per core virtual System Error physical pins.

## Parameters for ARMC1ProCT

### **cpuX.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **cpuX.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **cpuX.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **cpuX.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **cpuX.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **cpuX.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**cpuX.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**cpuX.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**cpuX.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**cpuX.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**cpuX.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**cpuX.l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: 0x100000

**cpuX.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**cpuX.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**cpuX.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**cpuX.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**cpuX.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**cpuX.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**cpuX.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**cpuX.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**cpuX.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**cpuX.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: true

**cpuX.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**cpuX.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**cpuX.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**cpuX.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**cpuX.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**cpuX.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**cpuX.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**AEND0\_DEFAULT**

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

Type: `uint64_t`

Default value: `0x0`

**AEND1\_DEFAULT**

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

Type: `uint64_t`

Default value: `0x0`

**AEND2\_DEFAULT**

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

Type: `uint64_t`

Default value: `0x0`

**AEND3\_DEFAULT**

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART0\_DEFAULT**

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART1\_DEFAULT**

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART2\_DEFAULT**

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART3\_DEFAULT**

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

Type: `uint64_t`

Default value: `0x0`

**BROADCASTATOMIC**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`

Default value: `true`

**BROADCASTCACHEMAINT**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

Type: `bool`

Default value: `false`



**BROADCASTOUTER**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

Type: `bool`

Default value: `false`

**BROADCASTPERSIST**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: `bool`

Default value: `true`

**CLUSTER\_ID**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the ManagerID64 of the core.

Type: `uint32_t`

Default value: `0x0`

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. This value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required.

Type: `uint8_t`

Default value: `1`

**CPUCFR**

Value of CPU Configuration Register.

Type: `uint64_t`

Default value: `0x0`

**DSU.cme.CMECFR**

Value of CMECFR\_EL1 fields (ECC, CHI) for all CMEs.

Type: `string`

Default value: `"{\\"ECC\\": 0, \\"CHI\\": 0}"`

**DSU.cme.mpmm\_accumulator\_multiplier**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator. value of `n` means the accumulator will use  $(n * \text{accumulator value})$  to calculate the mpmm threshold (MPMM). is provided as a fast model workaround to handle cases where execution of quantum in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

Type: `uint8_t`

Default value: `1`

**DSU.cme.power\_on\_by\_default**

If true, CME PPU's will initialize in Dynamic mode out of reset, enabling the CME to power itself on/off automatically as it is used.

Type: `bool`

Default value: `true`

**GICDISABLE**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`

Default value: `true`

**NUM\_CMES**

Defines how many CMEs are associated with each cluster. This parameter must be in sync with SVE.ScalableVectorExtension SME configuration.

Type: `uint8_t`

Default value: `0`

**NUM\_CORES**

Number of cores per cluster.

Type: `uint8_t`

Default value: `1`

**bus\_type**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

Type: `uint8_t`

Default value: 0

**core\_cache\_protection**

`core_cache_protection` can change `ERROFR`, `ERROPFGF` and `ERROPFGCTL` fields. Possible values are: -1:Not implemented (by default), 0:Disabled, 1:Enabled.

Type: `int8_t`

Default value: 1

**core\_power\_on\_by\_default**

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPUs which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

Type: `bool`

Default value: `false`

**cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: 0x1

**cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: 0x1

**dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**dcache-maintenance\_latency**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-miss\_latency**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-prefetch\_enabled**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

Type: `bool`

Default value: `false`

**dcache-read\_access\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-size**

L1 D-Cache size in bytes.

Type: `uint32_t`

Default value: 0x10000

### **dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **dcache-state\_modelled**

Set whether D-cache has stateful implementation.

Type: bool

Default value: false

### **dcache-write\_access\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **default\_opmode**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

Type: uint8\_t

Default value: 4

### **diagnostics**

Enable DynamIQ diagnostic messages.

Type: bool

Default value: `false`

### **`enable_simulation_performance_optimizations`**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of `stage12_tlb_size` parameter to 1024).

Type: `bool`

Default value: `true`

### **`ete.CLAIMTAGS`**

Number of claim tags.

Type: `uint8_t`

Default value: 4

### **`ete.ETE_REVISION`**

ETE revision: 0=ETEv1.0, 1=ETEv1.1, 2=ETEv1.2, 3=ETEv1.3.

Type: `uint8_t`

Default value: 1

### **`ete.MAX_INST_PER_Q`**

Maximum limit for the number of instructions implied by a Q element.

Type: `uint16_t`

Default value: 0x1

### **`ete.NumberOfRSPairs`**

Number of resource selector pairs.

Type: `uint8_t`

Default value: 8

### **`ete.PIDR_CM0D`**

TRCPIDR CM0D value.

Type: `uint8_t`

Default value: 0

**ete.PIDR\_REVAND**

TRCPIDR REVAND value.

Type: `uint8_t`

Default value: 0

**ete.PIDR\_REVISION**

TRCPIDR REVISION value.

Type: `uint8_t`

Default value: 0

**ete.Q\_CADENCE**

Number of instruction blocks traced between two Q elements.

Type: `uint16_t`

Default value: 0x1

**ete.RES0\_STATEFUL**

Whether **RES0** bits are stateful or **RAZ/WI**.

Type: `bool`

Default value: `false`

**ete.RETSTACK**

Return stack depth.

Type: `uint8_t`

Default value: 1

**ete.REVISION**

TRCIDR1 revision value.

Type: `uint8_t`

Default value: 1

**ete.SIM\_OVERFLOW\_GRANULARITY**

Number of instruction blocks in each granule, for simulated overflow.

Type: `uint32_t`

Default value: 0x64

**ete.SIM\_OVERFLOW\_PERCENTAGE**

Percentage of instruction blocks lost in each granule, for simulated overflow.

Type: `uint8_t`

Default value: `0`

**ete.SOURCE\_ADDRESS**

Allow generation of source address elements.

Type: `bool`

Default value: `false`

**ete.TRACE\_OUTPUT**

File to which to write trace byte stream.

Type: `string`

Default value: `N/A`

**ete.TRCSRSTA\_FORCED\_EXCEP**

TRCSRSTA value for a forcibly traced exception.

Type: `bool`

Default value: `false`

**ete.TSMARK**

Whether timestamp markers are supported.

Type: `bool`

Default value: `true`

**force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks**

Force MTE tag accesses to **RAZ/WI** and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

Type: `bool`

Default value: `false`



**force\_zero\_PSTATE\_PAN**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

**force\_zero\_mpam\_partid\_and\_pmg**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

Type: `uint8_t`

Default value: `0`

**has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

**has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

**has\_enhanced\_pan**

Implements Armv8.7 Enhanced PAN feature (FEAT\_PAN3) values of this parameter are:- 1, feature is implemented if ARMv8.7 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: `2`

**has\_ete**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (-plugin or -P).

Type: `bool`

Default value: `false`

**has\_large\_va**

Implement support for the extended 52-bit virtual addresses from ARMv8.2 (FEAT\_LVA). values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.2 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

**has\_peripheral\_port**

If true, an additional AXI peripheral port is configured.

Type: `bool`

Default value: `false`

**icache-hit\_latency**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

**icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-size**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: `0x10000`

**icache-state\_modelled**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

**instruction\_tlb\_size**

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

Type: `uint32_t`

Default value: `0x0`

**invalidate\_code\_cache\_on\_icache\_cmo**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: `uint8_t`

Default value: 0

### **l3cache-has\_mpam**

L3 Cache has MPAM support.

Type: `bool`

Default value: `true`

### **l3cache-hit\_latency**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **l3cache-maintenance\_latency**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **l3cache-miss\_latency**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **l3cache-mpamf.max\_partid\_ns**

Maximum value of non-secure PARTID supported for DSU L3Cache. Options are 7 or 63.

Type: `uint8_t`

Default value: 63

### **l3cache-read\_access\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size `l3cache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **l3cache-read\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l3cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **l3cache-size**

L3 Cache size in bytes.

Type: `uint32_t`

Default value: 0x80000

### **l3cache-snoop\_data\_transfer\_latency**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **l3cache-snoop\_issue\_latency**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **l3cache-ways**

L3 Cache number of ways (sets are implicit from size).

Type: `uint16_t`

Default value: 16

### **l3cache-write\_access\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size `l3cache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-write_latency`**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l3cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`memory_tagging_support_level`**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

Type: `uint8_t`

Default value: 3

### **`mpmm_accumulator_multiplier`**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator.value of `n` means the accumulator will use (`n * accumulator value`) to calculate the mpmm threshold (MPMM).is provided as a fast model workaround to handle cases where execution of quantum in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

Type: `uint8_t`

Default value: 1

### **`num_acp`**

Number of ACP ports.

Type: `uint8_t`

Default value: 0

### **`pmu-num_counters`**

Number of PMU counters implemented.

Type: `uint8_t`

Default value: 6

**ptw\_latency**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

**scu\_cache\_protection**

SCU-L3 is configured with ECC if true.

Type: `bool`

Default value: `true`

**stage12\_tlb\_size**

Number of stage1+2 tlb entries. Valid values are 0 or  $\geq 4$ .

Type: `uint32_t`

Default value: `0x80`

**tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

**tlbi\_stall\_enabled**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: `bool`

Default value: `false`

**treat\_PAC\_as\_NOP**

Non-architecture parameter. Treat Pointer Authentication as **NOP**. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are **NOP**. Branch, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

Type: `bool`

Default value: `false`

**walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

3.38 ARMC1ProCT\_C1UltraCT

Defined in LISA/ARMC1ProCT\_C1UltraCT.lisa.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
C1-Pro r0p0	Preliminary support
C1Ultra r0p0	Preliminary support

For an explanation of the quality levels, see [Quality level definitions](#).

Changes in 11.31.15

The following parameters were added:

- num\_acp
- subcluster0.has\_delayed\_dbgreg
- subcluster0.has\_delayed\_sysreg
- subcluster1.has\_delayed\_dbgreg
- subcluster1.has\_delayed\_sysreg
- subcluster1.mpam\_has\_altsp

About ARMC1ProCT\_C1UltraCT

The number of cores in each subcluster is configurable using the following parameters:

**subcluster0.NUM\_CORES**

Possible values are 1-13 (ARMC1ProCT).

**subcluster1.NUM\_CORES**

Possible values are 1-13 (ARMC1UltraCT).

The total number of cores in the cluster cannot exceed 14.

Port arrays are expanded to the width needed for the maximum number of cores in each subcluster. Use the following port array indexes:

- <port\_name>[0-12] for cores in subcluster0.
- <port\_name>[13-25] for cores in subcluster1.





All instances in the Master cross trigger matrix port array, `cti[26]` must be connected, regardless of the `NUM_CORES` value used.

Core-specific parameters have the following prefixes:

- `subcluster0.cpu0` to `subcluster0.cpu12` identify cores in `subcluster0`.
- `subcluster1.cpu0` to `subcluster1.cpu12` identify cores in `subcluster1`.

For information about the cores in this model, see:

- [ARMC1ProCT](#).
- [ARMC1UltraCT](#).

### Iris and MTI instances for ARMC1ProCT\_C1UltraCT

This model has the following Iris instances:

Name	Instance type
ARMC1ProCT_C1UltraCT	Cluster_ARM_C1_PRO_C1-ULTRA_Heterogeneous
ARMC1ProCT_C1UltraCT.AMU	PVBusLogger
ARMC1ProCT_C1UltraCT.AMU.mapper	PVBusMapper
ARMC1ProCT_C1UltraCT.DAP	PVBusLogger
ARMC1ProCT_C1UltraCT.DAP.mapper	PVBusMapper
ARMC1ProCT_C1UltraCT.DSU	C1-DSU
ARMC1ProCT_C1UltraCT.DSU.PPU_cluster	PPUv1
ARMC1ProCT_C1UltraCT.DSU.PPU_cluster.busslave	PVBusSlave
ARMC1ProCT_C1UltraCT.DSU.PPU_coreZ (where Z = 0-1)	PPUv1
ARMC1ProCT_C1UltraCT.DSU.PPU_coreZ.busslave (where Z = 0-1)	PVBusSlave
ARMC1ProCT_C1UltraCT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMC1ProCT_C1UltraCT.DSU.mpam_busslave	PVBusSlave
ARMC1ProCT_C1UltraCT.DSU.shared_cache	PVCache
ARMC1ProCT_C1UltraCT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMC1ProCT_C1UltraCT.DSU.shared_cache.upstream[Z] (where Z = 0-6)	PVBusSlave
ARMC1ProCT_C1UltraCT.DSU.utility_slave[0]	PVBusSlave
ARMC1ProCT_C1UltraCT.MMAP	PVBusLogger
ARMC1ProCT_C1UltraCT.MMAP.mapper	PVBusMapper
ARMC1ProCT_C1UltraCT.RAS	PVBusLogger
ARMC1ProCT_C1UltraCT.RAS.mapper	PVBusMapper
ARMC1ProCT_C1UltraCT.cpuZ.debug_rom (where Z = 0-1)	debug_rom
ARMC1ProCT_C1UltraCT.ext_bus	PVBusLogger
ARMC1ProCT_C1UltraCT.ext_bus.mapper	PVBusMapper
ARMC1ProCT_C1UltraCT.gic_cpuif_decoder_cluster	GLCv3CPUInterfaceDecoder

Name	Instance type
ARMC1ProCT_C1UltraCT.global_debug_rom	debug_rom
ARMC1ProCT_C1UltraCT.secondary_debug_rom	debug_rom
ARMC1ProCT_C1UltraCT.subcluster0	Subcluster_ARM_C1-Pro
ARMC1ProCT_C1UltraCT.subcluster0.cpu0	ARM_C1-Pro
ARMC1ProCT_C1UltraCT.subclusterZ.cpu0.UTLB (where Z = 0-1)	TLB
ARMC1ProCT_C1UltraCT.subclusterZ.cpuU.dtlb (where Z = 0-1; U = 0-1)	TLB
ARMC1ProCT_C1UltraCT.subclusterZ.cpu0.l1dcache (where Z = 0-1)	PVCache
ARMC1ProCT_C1UltraCT.subclusterZ.cpu0.l1dcache.downstream[0].pvbusmaster (where Z = 0-1)	PVBusMaster
ARMC1ProCT_C1UltraCT.subclusterZ.cpu0.l1dcache.upstream[0] (where Z = 0-1)	PVBusSlave
ARMC1ProCT_C1UltraCT.subclusterZ.cpu0.l1icache (where Z = 0-1)	PVCache
ARMC1ProCT_C1UltraCT.subclusterZ.cpu0.l1icache.downstream[0].pvbusmaster (where Z = 0-1)	PVBusMaster
ARMC1ProCT_C1UltraCT.subclusterZ.cpu0.l1icache.upstream[0] (where Z = 0-1)	PVBusSlave
ARMC1ProCT_C1UltraCT.subclusterZ.cpu0.l2cache (where Z = 0-1)	PVCache
ARMC1ProCT_C1UltraCT.subclusterZ.cpu0.l2cache.downstream[0].pvbusmaster (where Z = 0-1)	PVBusMaster
ARMC1ProCT_C1UltraCT.subclusterZ.cpu0.l2cache.upstream[W] (where Z = 0-1; W = 0-1)	PVBusSlave
ARMC1ProCT_C1UltraCT.subclusterZ.sve (where Z = 0-1)	ScalableVectorExtension
ARMC1ProCT_C1UltraCT.subcluster1	Subcluster_ARM_C1-Ultra
ARMC1ProCT_C1UltraCT.subcluster1.cpu0	ARM_C1-Ultra

This model has the following MTI trace components:

Name	Component type
ARMC1ProCT_C1UltraCT.AMU	PVBusLogger
ARMC1ProCT_C1UltraCT.AMU.mapper	PVBusMapper
ARMC1ProCT_C1UltraCT.DAP	PVBusLogger
ARMC1ProCT_C1UltraCT.DAP.mapper	PVBusMapper
ARMC1ProCT_C1UltraCT.DSU	C1-DSU
ARMC1ProCT_C1UltraCT.DSU.PPU_cluster	PPUv1
ARMC1ProCT_C1UltraCT.DSU.PPU_cluster.busslave	PVBusSlave
ARMC1ProCT_C1UltraCT.DSU.PPU_coreZ (where Z = 0-1)	PPUv1
ARMC1ProCT_C1UltraCT.DSU.PPU_coreZ.busslave (where Z = 0-1)	PVBusSlave
ARMC1ProCT_C1UltraCT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMC1ProCT_C1UltraCT.DSU.mpam_busslave	PVBusSlave
ARMC1ProCT_C1UltraCT.DSU.shared_cache	PVCache
ARMC1ProCT_C1UltraCT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMC1ProCT_C1UltraCT.DSU.shared_cache.upstream[Z] (where Z = 0-6)	PVBusSlave
ARMC1ProCT_C1UltraCT.DSU.utility_slave[0]	PVBusSlave

Name	Component type
ARMC1ProCT_C1UltraCT.MMAP	PVBusLogger
ARMC1ProCT_C1UltraCT.MMAP.mapper	PVBusMapper
ARMC1ProCT_C1UltraCT.RAS	PVBusLogger
ARMC1ProCT_C1UltraCT.RAS.mapper	PVBusMapper
ARMC1ProCT_C1UltraCT.ext_bus	PVBusLogger
ARMC1ProCT_C1UltraCT.ext_bus.mapper	PVBusMapper
ARMC1ProCT_C1UltraCT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMC1ProCT_C1UltraCT.subcluster0.cpu0	ARM_C1-Pro
ARMC1ProCT_C1UltraCT.subclusterZ.cpu0.UTLB (where Z = 0-1)	TLB
ARMC1ProCT_C1UltraCT.subclusterZ.cpu0.l1dcache (where Z = 0-1)	PVCache
ARMC1ProCT_C1UltraCT.subclusterZ.cpu0.l1dcache.downstream[0].pvbusmaster (where Z = 0-1)	PVBusMaster
ARMC1ProCT_C1UltraCT.subclusterZ.cpu0.l1dcache.upstream[0] (where Z = 0-1)	PVBusSlave
ARMC1ProCT_C1UltraCT.subclusterZ.cpu0.l1icache (where Z = 0-1)	PVCache
ARMC1ProCT_C1UltraCT.subclusterZ.cpu0.l1icache.downstream[0].pvbusmaster (where Z = 0-1)	PVBusMaster
ARMC1ProCT_C1UltraCT.subclusterZ.cpu0.l1icache.upstream[0] (where Z = 0-1)	PVBusSlave
ARMC1ProCT_C1UltraCT.subclusterZ.cpu0.l2cache (where Z = 0-1)	PVCache
ARMC1ProCT_C1UltraCT.subclusterZ.cpu0.l2cache.downstream[0].pvbusmaster (where Z = 0-1)	PVBusMaster
ARMC1ProCT_C1UltraCT.subclusterZ.cpu0.l2cache.upstream[W] (where Z = 0-1; W = 0-1)	PVBusSlave
ARMC1ProCT_C1UltraCT.subcluster1.cpu0	ARM_C1-Ultra

### Ports for ARMC1ProCT\_C1UltraCT

Port	Direction	Protocol	Description
acp_s	slave	PVBus	AXI ACP subordinate port.
AEND0MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND2MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND3MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART2MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	slave	Signal	CHI defined pins.

Port	Direction	Protocol	Description
broadcastcachemaint	slave	Signal	ACE defined pins.
broadcastrouter	slave	Signal	ACE defined pins.
broadcastpersist	slave	Signal	CHI defined pins.
cfgend	slave	Signal	This signal is for EE bit initialisation.
cfgsdisable	slave	Signal	This signal disables write access to some secure Interrupt Controller registers.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcs_m_pchannel	master	PChannel	Cluster PCSM signal
cluster_powerdown_out	master	Signal	DEPRECATED - An output from PPU that informs thermal controller of the cluster power info
cluster_ppu_hw_stat	master	Value	Cluster PPU power state
clusterid	slave	Value	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set ManagerID64 of the core.
clusterpmuirq	master	Signal	DynamiQ pmu irq
cme_pcs_m_pchannel	master	PChannel	Cluster PCSM signal
cmeerrirq	master	Signal	RAS cme err irq
cmefaultirq	master	Signal	RAS cme fault irq
CNTHPIRQ	master	Signal	Timer signals to SOC.
CNTHPSIRQ	master	Signal	Timer signals to SOC
CNTHVIRQ	master	Signal	Timer signals to SOC.
CNTHVSIRQ	master	Signal	Timer signals to SOC
CNTPNSIRQ	master	Signal	Timer signals to SOC.
CNTPSIRQ	master	Signal	Timer signals to SOC.
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.
CNTVIRQ	master	Signal	Timer signals to SOC.
commirq	master	Signal	Interrupt signal from debug communications channel.
core_clk_in	slave	ClockSignal	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
core_pcs_m_pchannel	master	PChannel	Core PCSM signals
core_powerdown_out	master	Signal	DEPRECATED - An output from PPU that informs thermal controller of the core power info
core_ppu_hw_stat	master	Value	Notify the power state change inside the PPU.

Port	Direction	Protocol	Description
coreerrirq	master	Signal	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq	master	Signal	Fault indicator for a detected 1 or 2 bit ECC error
coreinstrrun	master	Signal	Core Running State
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
ctidbgirq	master	Signal	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	slave	Signal	External debug interface.
dbgnopwrdown	master	Signal	No power-down request.
dbgppwrupreq	master	Signal	Debug power up request.
defaultMP	slave	Signal	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main manager interface ports.
dev_debug_s	slave	PVBus	External debug interface.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.
gicreset	master	Signal	An output from PPU that can be used to reset GICCLK domain components
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	slave	PVBus	External debug interface.
pmbirq	master	Signal	Interrupt signal from the statistical profiling unit.
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	master	Signal	PPU cluster interrupt.
ppu_cluster_isolate	master	Signal	PPU Cluster Domain isolation control.
ppu_cluster_wakerequest	slave	Signal	PPU cluster wake request signal.
ppu_cme_irq	master	Signal	PPU CME interrupt
ppu_core_irq	master	Signal	PPU core interrupt.
ppu_core_isolate	master	Signal	PPU Core Domain isolation control.
ppu_core_wakerequest	slave	Signal	PPU core wake request signal.
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	master	PVBus	The core will generate bus requests on this port.
pvbus_periph_m	master	PVBus	The core can generate peripheral bus request on this port.
rei	slave	Signal	Per core RAM Error Interrupt.
reset	slave	Signal	Reset signal to cluster.
rvbaraddr	slave	Value_64	Reset vector base address.
sci_m	master	SystemCoherencyInterface	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain

Port	Direction	Protocol	Description
sei	slave	Signal	Per core System Error physical pins.
spiden	slave	Signal	External debug interface.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq	master	Signal	Interrupt signal from the trace buffer unit.
utility_bus_s	slave	PVBus	Utility bus subordinate
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfiq	slave	Signal	Virtualised FIQ.
virq	slave	Signal	Virtualised IRQ.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Per core virtual System Error physical pins.

## Parameters for ARMC1ProCT\_C1UltraCT

### AEND0\_DEFAULT

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

Type: uint64\_t

Default value: 0x0

### AEND1\_DEFAULT

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

Type: uint64\_t

Default value: 0x0

### AEND2\_DEFAULT

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

Type: uint64\_t

Default value: 0x0

### AEND3\_DEFAULT

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

Type: uint64\_t

Default value: 0x0

#### **ASTART0\_DEFAULT**

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

Type: uint64\_t

Default value: 0x0

#### **ASTART1\_DEFAULT**

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

Type: uint64\_t

Default value: 0x0

#### **ASTART2\_DEFAULT**

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

Type: uint64\_t

Default value: 0x0

#### **ASTART3\_DEFAULT**

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

Type: uint64\_t

Default value: 0x0

#### **BROADCASTATOMIC**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: bool

Default value: true

#### **BROADCASTCACHEMAINT**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

Type: bool

Default value: `false`

### **BROADCASTOUTER**

Enable broadcasting of Outer Shareable transactions. The `broadcastouter` signal will override this value if used.

Type: `bool`

Default value: `false`

### **BROADCASTPERSIST**

Enable broadcasting of cache clean to the point of persistence operations. The `broadcastpersist` signal will override this value if used.

Type: `bool`

Default value: `true`

### **CLUSTER\_ID**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the ManagerID64 of the core.

Type: `uint32_t`

Default value: `0x0`

### **CMO\_broadcast\_when\_cache\_state\_modelling\_disabled**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. This value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required.

Type: `uint8_t`

Default value: `1`

### **GICDISABLE**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`



Default value: `true`

### **NUM\_CMES**

Defines how many CMEs are associated with each cluster. This parameter must be in sync with SVE.ScalableVectorExtension SME configuration.

Type: `uint8_t`

Default value: 0

### **bus\_type**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

Type: `uint8_t`

Default value: 0

### **core\_power\_on\_by\_default**

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPUs which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

Type: `bool`

Default value: `false`

### **dcache-state\_modelled**

Set whether D-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **default\_opmode**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

Type: `uint8_t`

Default value: 4

### **diagnostics**

Enable DynamIQ diagnostic messages.

Type: `bool`

Default value: `false`

**enable\_simulation\_performance\_optimizations**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

Type: `bool`

Default value: `true`

**has\_peripheral\_port**

If true, an additional AXI peripheral port is configured.

Type: `bool`

Default value: `false`

**icache-state\_modelled**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

**l3cache-has\_mpam**

L3 Cache has MPAM support.

Type: `bool`

Default value: `true`

**l3cache-hit\_latency**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-maintenance\_latency**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-miss\_latency**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**l3cache-mpamf.max\_partid\_ns**

Maximum value of non-secure PARTID supported for DSU L3Cache. Options are 7 or 63.

Type: uint8\_t

Default value: 63

**l3cache-read\_access\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size l3cache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**l3cache-read\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. l3cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**l3cache-size**

L3 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**l3cache-snoop\_data\_transfer\_latency**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **l3cache-snoop\_issue\_latency**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **l3cache-ways**

L3 Cache number of ways (sets are implicit from size).

Type: uint16\_t

Default value: 16

### **l3cache-write\_access\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size l3cache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **l3cache-write\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. l3cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **mpmm\_accumulator\_multiplier**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator.value of n means the accumulator will use (n \* accumulator value) to calculate the mpmm threshold (MPMM).is provided as a fast model workaround to handle cases where execution of quantum in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

Type: uint8\_t

Default value: 1

**num\_acp**

Number of ACP ports.

Type: `uint8_t`

Default value: 0

**scu\_cache\_protection**

SCU-L3 is configured with ECC if true.

Type: `bool`

Default value: `true`

**subcluster0.CPUCFR**

Value of CPU Configuration Register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.NUM\_CORES**

Number of cores in the subcluster. Total number of cores in cluster may not exceed 14.

Type: `uint8_t`

Default value: 1

**subcluster0.core\_cache\_protection**

`core_cache_protection` can change `ERROFR`, `ERROPFGF` and `ERROPFGCTL` fields. Possible values are: -1:Not implemented (by default), 0:Disabled, 1:Enabled.

Type: `int8_t`

Default value: 1

**subcluster0.cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**subcluster0.cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: 0x1

**subcluster0.cpu0.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: false

**subcluster0.cpu0.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: bool

Default value: false

**subcluster0.cpu0.CRYPTODISABLE**

Disable cryptographic features.

Type: bool

Default value: false

**subcluster0.cpu0.RVBARADDR**

Value of RVBAR\_ELx register.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter trace\_special\_hlt\_imm16.

Type: bool

Default value: false

**subcluster0.cpu0.force-fpsid**

Override the FPSID value.

Type: bool

Default value: true

**subcluster0.cpu0.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x100000

**subcluster0.cpu0.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu0.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t



Default value: 0

**subcluster0.cpu0.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu0.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu0.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu0.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster0.cpu0.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster0.cpu0.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster0.cpu0.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster0.cpu0.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: true

**subcluster0.cpu0.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu0.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster0.cpu0.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu0.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu0.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster0.cpu0.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster0.cpu1.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster0.cpu1.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster0.cpu1.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster0.cpu1.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu1.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu1.force-fpsid`**

Override the FPSID value.

Type: `bool`

Default value: `true`

### **`subcluster0.cpu1.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu1.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu1.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu1.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu1.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to

the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu1.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x100000`

#### **`subcluster0.cpu1.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu1.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu1.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu1.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu1.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu1.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu1.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu1.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu1.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu1.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster0.cpu1.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster0.cpu1.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu1.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu1.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu1.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu1.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu1.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu1.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: 0xf000000

**subcluster0.cpu1.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu1.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster0.cpu1.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: true

**subcluster0.cpu10.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: false

**subcluster0.cpu10.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: bool

Default value: false

**subcluster0.cpu10.CRYPTODISABLE**

Disable cryptographic features.

Type: bool

Default value: false



**subcluster0.cpu10.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu10.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu10.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster0.cpu10.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu10.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu10.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu10.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu10.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu10.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x100000

**subcluster0.cpu10.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu10.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu10.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu10.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu10.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

### **`subcluster0.cpu10.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

### **`subcluster0.cpu10.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu10.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu10.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu10.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster0.cpu10.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster0.cpu10.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster0.cpu10.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster0.cpu10.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu10.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: 0x0

**subcluster0.cpu10.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: 0xf000000

**subcluster0.cpu10.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu10.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu10.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu10.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster0.cpu10.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster0.cpu11.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster0.cpu11.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster0.cpu11.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster0.cpu11.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu11.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu11.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster0.cpu11.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu11.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu11.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu11.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu11.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu11.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x100000`

### **`subcluster0.cpu11.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu11.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu11.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu11.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu11.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu11.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu11.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000



**subcluster0.cpu11.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu11.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu11.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster0.cpu11.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster0.cpu11.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu11.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu11.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu11.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu11.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu11.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster0.cpu11.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu11.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu11.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster0.cpu11.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster0.cpu12.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu12.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu12.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu12.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu12.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu12.force-fpsid`**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster0.cpu12.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu12.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu12.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu12.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu12.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu12.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x100000

**subcluster0.cpu12.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu12.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu12.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu12.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu12.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu12.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu12.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster0.cpu12.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster0.cpu12.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: 0x123456

**subcluster0.cpu12.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster0.cpu12.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster0.cpu12.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster0.cpu12.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster0.cpu12.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: true

**subcluster0.cpu12.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu12.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu12.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster0.cpu12.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu12.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu12.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster0.cpu12.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster0.cpu2.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster0.cpu2.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster0.cpu2.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster0.cpu2.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu2.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.



Type: `bool`

Default value: `false`

### **`subcluster0.cpu2.force-fpsid`**

Override the FPSID value.

Type: `bool`

Default value: `true`

### **`subcluster0.cpu2.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu2.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu2.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu2.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu2.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to

the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu2.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x100000`

#### **`subcluster0.cpu2.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu2.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu2.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu2.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu2.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu2.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu2.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu2.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu2.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu2.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster0.cpu2.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster0.cpu2.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu2.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu2.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu2.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu2.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu2.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu2.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: 0xf000000

**subcluster0.cpu2.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu2.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster0.cpu2.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: true

**subcluster0.cpu3.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: false

**subcluster0.cpu3.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: bool

Default value: false

**subcluster0.cpu3.CRYPTODISABLE**

Disable cryptographic features.

Type: bool

Default value: false

**subcluster0.cpu3.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu3.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu3.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster0.cpu3.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu3.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu3.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu3.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x100000

**subcluster0.cpu3.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu3.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu3.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

### **`subcluster0.cpu3.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

### **`subcluster0.cpu3.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu3.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu3.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`



**subcluster0.cpu3.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster0.cpu3.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster0.cpu3.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster0.cpu3.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster0.cpu3.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu3.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: 0x0

**subcluster0.cpu3.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: 0xf000000

**subcluster0.cpu3.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu3.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu3.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu3.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster0.cpu3.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster0.cpu4.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster0.cpu4.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster0.cpu4.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster0.cpu4.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu4.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu4.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster0.cpu4.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu4.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu4.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu4.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu4.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu4.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x100000`

### **`subcluster0.cpu4.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu4.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu4.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu4.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu4.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu4.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu4.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster0.cpu4.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster0.cpu4.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu4.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu4.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu4.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu4.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster0.cpu4.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu4.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu4.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster0.cpu4.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster0.cpu5.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu5.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu5.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu5.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu5.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu5.force-fpsid`**

Override the FPSID value.

Type: `bool`

Default value: `true`



**subcluster0.cpu5.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x100000

#### **subcluster0.cpu5.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu5.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu5.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu5.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu5.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

#### **subcluster0.cpu5.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu5.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu5.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu5.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu5.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster0.cpu5.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster0.cpu5.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster0.cpu5.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster0.cpu5.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: true

**subcluster0.cpu5.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu5.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster0.cpu5.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu5.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu5.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster0.cpu5.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster0.cpu6.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster0.cpu6.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster0.cpu6.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster0.cpu6.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu6.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu6.force-fpsid`**

Override the FPSID value.

Type: `bool`

Default value: `true`

### **`subcluster0.cpu6.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu6.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu6.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu6.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu6.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to

the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu6.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x100000`

#### **`subcluster0.cpu6.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu6.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu6.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu6.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu6.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu6.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu6.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu6.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu6.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu6.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster0.cpu6.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t



Default value: 171

**subcluster0.cpu6.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu6.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu6.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu6.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu6.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu6.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu6.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: 0xf000000

**subcluster0.cpu6.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu6.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster0.cpu6.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: true

**subcluster0.cpu7.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: false

**subcluster0.cpu7.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: bool

Default value: false

**subcluster0.cpu7.CRYPTODISABLE**

Disable cryptographic features.

Type: bool

Default value: false

**subcluster0.cpu7.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu7.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu7.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster0.cpu7.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu7.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu7.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu7.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x100000

**subcluster0.cpu7.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu7.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu7.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

### **`subcluster0.cpu7.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

### **`subcluster0.cpu7.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu7.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu7.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu7.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster0.cpu7.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster0.cpu7.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster0.cpu7.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster0.cpu7.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu7.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu7.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu7.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu7.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu7.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu7.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster0.cpu7.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster0.cpu8.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster0.cpu8.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster0.cpu8.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster0.cpu8.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu8.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu8.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster0.cpu8.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu8.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.



Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu8.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu8.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu8.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu8.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x100000`

### **`subcluster0.cpu8.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu8.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu8.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu8.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu8.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu8.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu8.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster0.cpu8.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster0.cpu8.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu8.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu8.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu8.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu8.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster0.cpu8.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu8.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu8.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster0.cpu8.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu9.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu9.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu9.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu9.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu9.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu9.force-fpsid`**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster0.cpu9.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu9.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu9.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu9.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu9.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu9.l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: 0x100000

**subcluster0.cpu9.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu9.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu9.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster0.cpu9.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster0.cpu9.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: 0x123456

**subcluster0.cpu9.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster0.cpu9.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster0.cpu9.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster0.cpu9.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`



Default value: N/A

**subcluster0.cpu9.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: true

**subcluster0.cpu9.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu9.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster0.cpu9.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu9.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu9.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster0.cpu9.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster0.dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.dcache-maintenance\_latency**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.dcache-miss\_latency**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.dcache-prefetch\_enabled**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

Type: `bool`

Default value: `false`

**subcluster0.dcache-read\_access\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.dcache-size**

L1 D-Cache size in bytes.

Type: uint32\_t

Default value: 0x10000

**subcluster0.dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.dcache-write\_access\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.ete.CLAIMTAGS**

Number of claim tags.

Type: uint8\_t

Default value: 4

**subcluster0.ete.ETE\_REVISION**

ETE revision: 0=ETEv1.0, 1=ETEv1.1, 2=ETEv1.2, 3=ETEv1.3.

Type: uint8\_t

Default value: 1

**subcluster0.ete.MAX\_INST\_PER\_Q**

Maximum limit for the number of instructions implied by a Q element.

Type: uint16\_t

Default value: 0x1

**subcluster0.ete.NumberOfRSPairs**

Number of resource selector pairs.

Type: uint8\_t

Default value: 8

**subcluster0.ete.PIDR\_CM0D**

TRCPIDR CM0D value.

Type: uint8\_t

Default value: 0

**subcluster0.ete.PIDR\_REVAND**

TRCPIDR REVAND value.

Type: `uint8_t`

Default value: 0

#### **subcluster0.ete.PIDR\_REVISION**

TRCPIDR REVISION value.

Type: `uint8_t`

Default value: 0

#### **subcluster0.ete.Q\_CADENCE**

Number of instruction blocks traced between two Q elements.

Type: `uint16_t`

Default value: 0x1

#### **subcluster0.ete.RES0\_STATEFUL**

Whether **RES0** bits are stateful or **RAZ/WI**.

Type: `bool`

Default value: `false`

#### **subcluster0.ete.RETSTACK**

Return stack depth.

Type: `uint8_t`

Default value: 1

#### **subcluster0.ete.REVISION**

TRCIDR1 revision value.

Type: `uint8_t`

Default value: 1

#### **subcluster0.ete.SIM\_OVERFLOW\_GRANULARITY**

Number of instruction blocks in each granule, for simulated overflow.

Type: `uint32_t`

Default value: 0x64

#### **subcluster0.ete.SIM\_OVERFLOW\_PERCENTAGE**

Percentage of instruction blocks lost in each granule, for simulated overflow.

Type: `uint8_t`

Default value: 0

#### **`subcluster0.ete.SOURCE_ADDRESS`**

Allow generation of source address elements.

Type: `bool`

Default value: `false`

#### **`subcluster0.ete.TRACE_OUTPUT`**

File to which to write trace byte stream.

Type: `string`

Default value: N/A

#### **`subcluster0.ete.TRCRSRTA_FORCED_EXCEP`**

TRCRSR.TA value for a forcibly traced exception.

Type: `bool`

Default value: `false`

#### **`subcluster0.ete.TSMARK`**

Whether timestamp markers are supported.

Type: `bool`

Default value: `true`

#### **`subcluster0.force_mte_tag_access_razwi_and_ignore_tag_checks`**

Force MTE tag accesses to **RAZ/WI** and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

Type: `bool`

Default value: `false`

#### **`subcluster0.force_zero_PSTATE_PAN`**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

### **`subcluster0.force_zero_mpam_partid_and_pmg`**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAMO\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

Type: `uint8_t`

Default value: 0

### **`subcluster0.has_delayed_dbgreg`**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

### **`subcluster0.has_delayed_sysreg`**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

### **`subcluster0.has_enhanced_pan`**

Implements Armv8.7 Enhanced PAN feature (FEAT\_PAN3) values of this parameter are:- 1, feature is implemented if ARMv8.7 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 2

### **`subcluster0.has_ete`**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (-plugin or -P).

Type: `bool`

Default value: `false`

### **`subcluster0.has_large_va`**

Implement support for the extended 52-bit virtual addresses from ARMv8.2 (FEAT\_LVA). values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.2 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

#### **subcluster0.icache-hit\_latency**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

Type: bool

Default value: false

#### **subcluster0.icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to



the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.icache-size`**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: `0x10000`

#### **`subcluster0.instruction_tlb_size`**

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

Type: `uint32_t`

Default value: `0x0`

#### **`subcluster0.invalidate_code_cache_on_icache_cmo`**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: `uint8_t`

Default value: `0`

#### **`subcluster0.memory_tagging_support_level`**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

Type: `uint8_t`

Default value: `3`

#### **`subcluster0.pmu-num_counters`**

Number of PMU counters implemented.

Type: `uint8_t`

Default value: `6`

#### **`subcluster0.ptw_latency`**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.stage12_tlb_size`**

Number of stage1+2 tlb entries. Valid values are 0 or  $\geq 4$ .

Type: `uint32_t`

Default value: `0x80`

### **`subcluster0.tlb_latency`**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.tlbi_stall_enabled`**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: `bool`

Default value: `false`

### **`subcluster0.treat_PAC_as_NOP`**

Non-architecture parameter. Treat Pointer Authentication as **NOP**. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are **NOP**. Branch, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

Type: `bool`

Default value: `false`

### **`subcluster0.walk_cache_latency`**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.CPUCFR`**

Value of CPU Configuration Register.

Type: `uint64_t`

Default value: 0x1

**subcluster1.NUM\_CORES**

Number of cores in the subcluster. Total number of cores in cluster may not exceed 14.

Type: uint8\_t

Default value: 1

**subcluster1.cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: uint32\_t

Default value: 0x1

**subcluster1.cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: uint32\_t

Default value: 0x1

**subcluster1.cpu0.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: false

**subcluster1.cpu0.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: bool

Default value: false

**subcluster1.cpu0.CRYPTODISABLE**

Disable cryptographic features.

Type: bool

Default value: false

**subcluster1.cpu0.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu0.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu0.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu0.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu0.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu0.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu0.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to

the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu0.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

#### **`subcluster1.cpu0.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu0.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu0.l2cache-ways`**

L2 Cache number of ways (sets are implicit from size).

Type: `uint8_t`

Default value: `8`

#### **`subcluster1.cpu0.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu0.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu0.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu0.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu0.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu0.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu0.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu0.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

### **`subcluster1.cpu0.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

### **`subcluster1.cpu0.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **`subcluster1.cpu0.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **`subcluster1.cpu0.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster1.cpu0.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu0.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster1.cpu0.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster1.cpu0.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster1.cpu0.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu0.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu0.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster1.cpu1.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster1.cpu1.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`



Default value: `false`

**subcluster1.cpu1.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu1.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu1.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu1.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu1.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu1.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu1.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu1.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu1.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster1.cpu1.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu1.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu1.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster1.cpu1.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu1.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu1.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu1.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu1.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu1.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu1.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu1.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster1.cpu1.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster1.cpu1.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu1.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu1.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu1.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu1.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu1.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu1.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu1.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu1.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu1.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster1.cpu10.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu10.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster1.cpu10.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu10.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu10.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu10.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu10.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu10.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu10.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu10.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu10.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

### **`subcluster1.cpu10.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu10.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu10.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster1.cpu10.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu10.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu10.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu10.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0



**subcluster1.cpu10.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu10.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu10.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu10.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster1.cpu10.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster1.cpu10.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu10.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu10.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu10.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu10.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu10.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu10.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu10.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu10.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu10.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster1.cpu11.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster1.cpu11.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu11.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu11.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu11.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu11.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu11.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu11.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu11.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu11.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu11.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

#### **subcluster1.cpu11.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu11.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu11.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

#### **subcluster1.cpu11.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu11.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu11.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu11.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu11.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu11.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu11.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu11.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster1.cpu11.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster1.cpu11.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster1.cpu11.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster1.cpu11.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu11.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu11.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu11.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu11.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu11.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu11.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu11.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster1.cpu12.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster1.cpu12.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu12.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu12.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`



**subcluster1.cpu12.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter trace\_special\_hlt\_imm16.

Type: bool

Default value: false

**subcluster1.cpu12.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu12.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu12.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu12.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu12.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu12.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

#### **subcluster1.cpu12.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu12.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu12.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

#### **subcluster1.cpu12.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu12.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu12.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu12.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu12.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu12.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu12.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu12.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster1.cpu12.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

### **`subcluster1.cpu12.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **`subcluster1.cpu12.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **`subcluster1.cpu12.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster1.cpu12.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu12.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster1.cpu12.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster1.cpu12.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

#### **`subcluster1.cpu12.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster1.cpu12.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu12.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster1.cpu2.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu2.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu2.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu2.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu2.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

#### **subcluster1.cpu2.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

#### **subcluster1.cpu2.l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: 0x80000

#### **subcluster1.cpu2.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

#### **subcluster1.cpu2.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

#### **subcluster1.cpu2.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: `uint8_t`

Default value: 8

#### **subcluster1.cpu2.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **subcluster1.cpu2.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster1.cpu2.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

### **subcluster1.cpu2.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

### **subcluster1.cpu2.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

### **subcluster1.cpu2.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

### **subcluster1.cpu2.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456



**subcluster1.cpu2.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster1.cpu2.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster1.cpu2.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster1.cpu2.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster1.cpu2.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu2.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: 0x0

**subcluster1.cpu2.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: 0xf000000

**subcluster1.cpu2.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu2.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu2.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu2.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu2.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster1.cpu3.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu3.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster1.cpu3.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu3.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu3.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu3.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu3.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu3.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

#### **`subcluster1.cpu3.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu3.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster1.cpu3.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu3.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu3.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu3.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu3.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu3.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster1.cpu3.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster1.cpu3.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu3.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu3.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu3.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu3.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster1.cpu3.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu3.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu3.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster1.cpu3.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster1.cpu4.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu4.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu4.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu4.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu4.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu4.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`



**subcluster1.cpu4.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu4.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu4.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu4.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu4.l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

**subcluster1.cpu4.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster1.cpu4.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu4.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster1.cpu4.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu4.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu4.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu4.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu4.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster1.cpu4.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster1.cpu4.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: 0x123456

**subcluster1.cpu4.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster1.cpu4.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster1.cpu4.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster1.cpu4.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster1.cpu4.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: true

**subcluster1.cpu4.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu4.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu4.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster1.cpu4.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu4.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu4.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu4.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster1.cpu5.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu5.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster1.cpu5.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu5.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu5.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu5.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu5.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu5.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu5.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu5.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu5.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster1.cpu5.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu5.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu5.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster1.cpu5.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu5.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu5.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu5.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu5.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu5.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu5.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu5.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster1.cpu5.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t



Default value: 171

**subcluster1.cpu5.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu5.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu5.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu5.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu5.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu5.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu5.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: 0xf000000

**subcluster1.cpu5.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu5.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster1.cpu5.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: true

**subcluster1.cpu6.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: false

**subcluster1.cpu6.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: bool

Default value: false

**subcluster1.cpu6.CRYPTODISABLE**

Disable cryptographic features.

Type: bool

Default value: false

**subcluster1.cpu6.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu6.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu6.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu6.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu6.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu6.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu6.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu6.l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

**subcluster1.cpu6.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu6.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu6.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: `uint8_t`

Default value: `8`

**subcluster1.cpu6.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu6.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu6.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu6.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu6.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu6.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu6.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu6.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**`subcluster1.cpu6.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**`subcluster1.cpu6.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**`subcluster1.cpu6.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**`subcluster1.cpu6.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**`subcluster1.cpu6.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: 0x0

**`subcluster1.cpu6.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: 0xf000000

**`subcluster1.cpu6.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster1.cpu6.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster1.cpu6.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu6.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu6.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster1.cpu7.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster1.cpu7.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu7.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu7.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu7.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu7.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu7.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu7.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`



**subcluster1.cpu7.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu7.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu7.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster1.cpu7.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu7.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu7.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster1.cpu7.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu7.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu7.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu7.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu7.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu7.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu7.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu7.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster1.cpu7.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster1.cpu7.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu7.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu7.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu7.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu7.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu7.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu7.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu7.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu7.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu7.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster1.cpu8.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu8.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster1.cpu8.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu8.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu8.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu8.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu8.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu8.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu8.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu8.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu8.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

### **`subcluster1.cpu8.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu8.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu8.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster1.cpu8.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu8.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu8.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu8.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu8.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu8.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu8.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu8.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster1.cpu8.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster1.cpu8.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu8.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`



**subcluster1.cpu8.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu8.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu8.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu8.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu8.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu8.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu8.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu8.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster1.cpu9.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster1.cpu9.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu9.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu9.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu9.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu9.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster1.cpu9.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster1.cpu9.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu9.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu9.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu9.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu9.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu9.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster1.cpu9.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster1.cpu9.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster1.cpu9.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster1.cpu9.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu9.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu9.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu9.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu9.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu9.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu9.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu9.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster1.dcache-hit_latency`**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.dcache-maintenance_latency`**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.dcache-miss_latency`**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.dcache-prefetch_enabled`**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

Type: `bool`

Default value: `false`

### **`subcluster1.dcache-read_access_latency`**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.dcache-read_latency`**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.dcache-size`**

L1 D-Cache size in bytes.

Type: `uint32_t`

Default value: `0x10000`

### **`subcluster1.dcache-snoop_data_transfer_latency`**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.dcache-write_access_latency`**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`



**subcluster1.dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.ecv\_support\_level**

Implement Enhanced Counter Virtualization feature from ARMv8.6. 0, Not supported. 1, fully supported without CNTPOFF. 2, fully supported with CNTPOFF (FEAT\_ECV).

Type: uint8\_t

Default value: 2

**subcluster1.ete.CLAIMTAGS**

Number of claim tags.

Type: uint8\_t

Default value: 4

**subcluster1.ete.ETE\_REVISION**

ETE revision: 0=ETEv1.0, 1=ETEv1.1, 2=ETEv1.2, 3=ETEv1.3.

Type: uint8\_t

Default value: 1

**subcluster1.ete.MAX\_INST\_PER\_Q**

Maximum limit for the number of instructions implied by a Q element.

Type: uint16\_t

Default value: 0x1

**subcluster1.ete.NumberOfRSPairs**

Number of resource selector pairs.

Type: uint8\_t

Default value: 8

**subcluster1.ete.PIDR\_CM0D**

TRCPIDR CM0D value.

Type: `uint8_t`

Default value: 0

#### **`subcluster1.ete.Q_CADENCE`**

Number of instruction blocks traced between two Q elements.

Type: `uint16_t`

Default value: 0x1

#### **`subcluster1.ete.RES0_STATEFUL`**

Whether **RES0** bits are stateful or **RAZ/WI**.

Type: `bool`

Default value: `false`

#### **`subcluster1.ete.RETSTACK`**

Return stack depth.

Type: `uint8_t`

Default value: 3

#### **`subcluster1.ete.SIM_OVERFLOW_GRANULARITY`**

Number of instruction blocks in each granule, for simulated overflow.

Type: `uint32_t`

Default value: 0x64

#### **`subcluster1.ete.SIM_OVERFLOW_PERCENTAGE`**

Percentage of instruction blocks lost in each granule, for simulated overflow.

Type: `uint8_t`

Default value: 0

#### **`subcluster1.ete.SOURCE_ADDRESS`**

Allow generation of source address elements.

Type: `bool`

Default value: `false`

#### **`subcluster1.ete.TRACE_OUTPUT`**

File to which to write trace byte stream.

Type: `string`

Default value: `N/A`

#### **`subcluster1.ete.TRCRSRTA_FORCED_EXCEP`**

TRCRSR.TA value for a forcibly traced exception.

Type: `bool`

Default value: `false`

#### **`subcluster1.ext_abort_so_write_ras_type`**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO , 4 = UER, 5 = CE.

Type: `uint8_t`

Default value: `2`

#### **`subcluster1.force_mte_tag_access_razwi_and_ignore_tag_checks`**

Force MTE tag accesses to **RAZ/WI** and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

Type: `bool`

Default value: `false`

#### **`subcluster1.force_zero_PSTATE_PAN`**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

#### **`subcluster1.force_zero_mpam_partid_and_pmg`**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAMO\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

Type: `uint8_t`

Default value: `0`

#### **`subcluster1.has_delayed_dbgreg`**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

### **`subcluster1.has_delayed_sysreg`**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

### **`subcluster1.has_enhanced_pan`**

Implements Armv8.7 Enhanced PAN feature (FEAT\_PAN3) values of this parameter are:- 1, feature is implemented if ARMv8.7 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 2

### **`subcluster1.has_ete`**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (-plugin or -P).

Type: `bool`

Default value: `false`

### **`subcluster1.has_mt_pmu_disable_feature`**

Implements Multi-threading PMU disable extension from ARMv8.6 (FEAT\_MTPMU). 0: FEAT\_MTPMU is disabled, 1: FEAT\_MTPMU is enabled if ARMv8.6 is implemented, 2: FEAT\_MTPMU is cherry-picked, 0xF: The feature is disabled and is represented by value 0xF in ID\_AA64DFR0\_EL1.MTPMU.

Type: `uint8_t`

Default value: 0

### **`subcluster1.has_statistical_profiling`**

Whether Statistical Based Profiling is implemented (FEAT\_SPE).

Type: `bool`

Default value: `true`

### **`subcluster1.has_v8_7_spe_inverted_filtering`**

Where FEAT\_SPEv1p2 is implemented, whether inverted filtering by events is implemented (represented by PMISDR.FnE).

Type: `bool`

Default value: `false`

### **subcluster1.has\_v8\_7\_spe\_previous\_branch\_target**

Where FEAT\_SPEv1p2 is implemented, whether the optional branch target feature is implemented (FEAT\_SPE\_PBT).

Type: `bool`

Default value: `false`

### **subcluster1.icache-hit\_latency**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **subcluster1.icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **subcluster1.icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **subcluster1.icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

### **subcluster1.icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and

intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.icache-read_latency`**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.icache-size`**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: `0x10000`

#### **`subcluster1.instruction_tlb_size`**

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

Type: `uint32_t`

Default value: `0x0`

#### **`subcluster1.invalidate_code_cache_on_icache_cmo`**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: `uint8_t`

Default value: `0`

#### **`subcluster1.memory_tagging_support_level`**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

Type: `uint8_t`

Default value: `3`

**subcluster1.mpam\_has\_altsp**

DEPRECATED: MPAMIDR\_EL1.HAS\_ALTSP is required when FEAT\_RME is implemented.

Type: `bool`

Default value: `false`

**subcluster1.mpamidr\_has\_force\_ns**

Whether MPAMIDR\_EL1.HAS\_FORCE\_NS bit is set or clear. values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.6 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

**subcluster1.mpamidr\_has\_sdeflt**

Whether MPAMIDR\_EL1.HAS\_SDEFLT bit is set or clear. values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.6 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

**subcluster1.mpamidr\_has\_tidr**

Whether MPAMIDR\_EL1.HAS\_TIDR bit is set or clear. values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.6 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

**subcluster1.pmu-num\_counters**

Number of PMU counters implemented.

Type: `uint8_t`

Default value: 31

**subcluster1.ptw\_latency**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

**subcluster1.stage12\_tlb\_size**

Number of stage1+2 tlb entries. Valid values are 0 or  $\geq 4$ .

Type: `uint32_t`

Default value: `0x80`

**subcluster1.tcr\_txsz\_undersize\_should\_fault**

If large VA is not supported, Whether undersized TxSZ value should generate translation fault.

Type: `bool`

Default value: `false`

**subcluster1.tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.tlbi\_stall\_enabled**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: `bool`

Default value: `false`

**subcluster1.treat\_PAC\_as\_NOP**

Non-architecture parameter. Treat Pointer Authentication as **NOP**. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are **NOP**. Branch, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

Type: `bool`

Default value: `false`

**subcluster1.walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`



## 3.39 ARMC1UltraCT

Defined in `LISA/ARMC1UltraCT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Preliminary support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following parameters were added:

- `has_delayed_dbgreg`
- `has_delayed_sysreg`
- `mpam_has_altsp`
- `num_acp`

### About ARMC1UltraCT

The following example platforms are available:

- `FVP_Base_C1-Ultra`
- `EVS_Dhrystone_C1-Ultrax1`
- `SVP_Base_C1-Ultrax1`

The following functionality is supported in this release:

- C1-SME2 is supported with the following limitations:
  - The arbitration and assignment logic is not supported.
  - RAS error handling is supported in the first CME only.
  - The C1-SME2 auxiliary AMU counters are not supported.
  - It is assumed that C1-SME2 units are operating in dynamic mode, with a minimum power mode of OFF. Accompanying PPU registers may be read or written to, but they do not determine the behavior of the CME. The PPU interrupt signal for the C1-SME2 is also not modeled or exposed at this time.
- Pilatus DSU support for C1-SME2.
- L2Cache is supported at the per-core level only.
- BROADCASTPERSIST pin is implemented.
- Optional peripheral port is supported.
- L3Cache partition is supported.
- Per-core clock is supported.

## Limitations

The following features are not yet supported, and will be added in a future release:

- BROADCASTCACHEMAINTPOU pin is not implemented.
- COREINSTRET signals are not implemented.
- MEM\_RET power mode is not supported.

The following features are not implemented:

- DynamIQ features that are negligible to the programmers' view simulation will not be implemented in the Fast Model.
- 256-bit wide output transactions will not be supported.
- Error correction/detection features will not be supported.
- Self-test features (MBIST) will not be supported.
- Latency configuration will not be supported.
- Snoop filtering will not be supported.
- Cache stashing capability will not be supported.

## Iris and MTI instances for ARMC1UltraCT

This model has the following Iris instances:

Name	Instance type
ARMC1UltraCT	Cluster_ARM_C1-Ultra
ARMC1UltraCT.AMU	PVBusLogger
ARMC1UltraCT.AMU.mapper	PVBusMapper
ARMC1UltraCT.DAP	PVBusLogger
ARMC1UltraCT.DAP.mapper	PVBusMapper
ARMC1UltraCT.DSU	C1-DSU
ARMC1UltraCT.DSU.PPU_cluster	PPUv1
ARMC1UltraCT.DSU.PPU_cluster.busslave	PVBusSlave
ARMC1UltraCT.DSU.PPU_core0	PPUv1
ARMC1UltraCT.DSU.PPU_core0.busslave	PVBusSlave
ARMC1UltraCT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMC1UltraCT.DSU.mpam_busslave	PVBusSlave
ARMC1UltraCT.DSU.shared_cache	PVCache
ARMC1UltraCT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMC1UltraCT.DSU.shared_cache.upstream[Y] (where Y = 0–4)	PVBusSlave
ARMC1UltraCT.DSU.utility_slave[0]	PVBusSlave
ARMC1UltraCT.MMAP	PVBusLogger
ARMC1UltraCT.MMAP.mapper	PVBusMapper
ARMC1UltraCT.RAS	PVBusLogger
ARMC1UltraCT.RAS.mapper	PVBusMapper

Name	Instance type
ARMC1UltraCT.cpu0	ARM_C1-Ultra
ARMC1UltraCT.cpu0.UTLB	TLB
ARMC1UltraCT.cpu0.debug_rom	debug_rom
ARMC1UltraCT.cpu0.dtlb	TLB
ARMC1UltraCT.cpu0.l1dcache	PVCache
ARMC1UltraCT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMC1UltraCT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMC1UltraCT.cpu0.l1licache	PVCache
ARMC1UltraCT.cpu0.l1licache.downstream[0].pvbusmaster	PVBusMaster
ARMC1UltraCT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMC1UltraCT.cpu0.l2cache	PVCache
ARMC1UltraCT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMC1UltraCT.cpu0.l2cache.upstream[U] (where U = 0-1)	PVBusSlave
ARMC1UltraCT.ext_bus	PVBusLogger
ARMC1UltraCT.ext_bus.mapper	PVBusMapper
ARMC1UltraCT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMC1UltraCT.global_debug_rom	debug_rom
ARMC1UltraCT.secondary_debug_rom	debug_rom
ARMC1UltraCT.sve	ScalableVectorExtension

This model has the following MTI trace components:

Name	Component type
ARMC1UltraCT.AMU	PVBusLogger
ARMC1UltraCT.AMU.mapper	PVBusMapper
ARMC1UltraCT.DAP	PVBusLogger
ARMC1UltraCT.DAP.mapper	PVBusMapper
ARMC1UltraCT.DSU	C1-DSU
ARMC1UltraCT.DSU.PPU_cluster	PPUv1
ARMC1UltraCT.DSU.PPU_cluster.busslave	PVBusSlave
ARMC1UltraCT.DSU.PPU_core0	PPUv1
ARMC1UltraCT.DSU.PPU_core0.busslave	PVBusSlave
ARMC1UltraCT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMC1UltraCT.DSU.mpam_busslave	PVBusSlave
ARMC1UltraCT.DSU.shared_cache	PVCache
ARMC1UltraCT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMC1UltraCT.DSU.shared_cache.upstream[Y] (where Y = 0-4)	PVBusSlave
ARMC1UltraCT.DSU.utility_slave[0]	PVBusSlave
ARMC1UltraCT.MMAP	PVBusLogger
ARMC1UltraCT.MMAP.mapper	PVBusMapper
ARMC1UltraCT.RAS	PVBusLogger

Name	Component type
ARMC1UltraCT.RAS.mapper	PVBusMapper
ARMC1UltraCT.cpu0	ARM_C1-Ultra
ARMC1UltraCT.cpu0.UTLB	TLB
ARMC1UltraCT.cpu0.l1dcache	PVCache
ARMC1UltraCT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMC1UltraCT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMC1UltraCT.cpu0.l1licache	PVCache
ARMC1UltraCT.cpu0.l1licache.downstream[0].pvbusmaster	PVBusMaster
ARMC1UltraCT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMC1UltraCT.cpu0.l2cache	PVCache
ARMC1UltraCT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMC1UltraCT.cpu0.l2cache.upstream[U] (where U = 0-1)	PVBusSlave
ARMC1UltraCT.ext_bus	PVBusLogger
ARMC1UltraCT.ext_bus.mapper	PVBusMapper
ARMC1UltraCT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

## Ports for ARMC1UltraCT

Port	Direction	Protocol	Description
acp_s	slave	PVBus	AXI ACP subordinate port.
AEND0MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND2MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND3MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART2MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	slave	Signal	CHI defined pins.
broadcastcachemaint	slave	Signal	ACE defined pins.
broadcastouter	slave	Signal	ACE defined pins.
broadcastpersist	slave	Signal	CHI defined pins.
cfgend	slave	Signal	This signal is for EE bit initialisation.
cfgsdisable	slave	Signal	This signal disables write access to some secure Interrupt Controller registers.

Port	Direction	Protocol	Description
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcs_m_pchannel	master	PChannel	Cluster PCSM signal
cluster_powerdown_out	master	Signal	DEPRECATED - An output from PPUs that informs thermal controller of the cluster power info
cluster_ppu_hw_stat	master	Value	Cluster PPU power state
clusterid	slave	Value	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set ManagerID64 of the core.
clusterpmuirq	master	Signal	DynamiQ pmu irq
cme_pcs_m_pchannel	master	PChannel	Cluster PCSM signal
cmeerriq	master	Signal	RAS cme err irq
cmefaultirq	master	Signal	RAS cme fault irq
CNTHPIRQ	master	Signal	Timer signals to SOC.
CNTHPSIRQ	master	Signal	Timer signals to SOC
CNTHVIRQ	master	Signal	Timer signals to SOC.
CNTHVSIRQ	master	Signal	Timer signals to SOC
CNTPNSIRQ	master	Signal	Timer signals to SOC.
CNTPSIRQ	master	Signal	Timer signals to SOC.
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.
CNTVIRQ	master	Signal	Timer signals to SOC.
commirq	master	Signal	Interrupt signal from debug communications channel.
core_clk_in	slave	ClockSignal	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
core_pcs_m_pchannel	master	PChannel	Core PCSM signals
core_powerdown_out	master	Signal	DEPRECATED - An output from PPUs that informs thermal controller of the core power info
core_ppu_hw_stat	master	Value	Notify the power state change inside the PPU.
coreerrirq	master	Signal	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq	master	Signal	Fault indicator for a detected 1 or 2 bit ECC error
coreinstrrun	master	Signal	Core Running State
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
ctidbgirq	master	Signal	Cross Trigger Interface (CTI) interrupt trigger output.

Port	Direction	Protocol	Description
dbgen	slave	Signal	External debug interface.
dbgnopwrdown	master	Signal	No power-down request.
dbgpwrupreq	master	Signal	Debug power up request.
defaultMP	slave	Signal	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main manager interface ports.
dev_debug_s	slave	PVBus	External debug interface.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.
gicreset	master	Signal	An output from PPU that can be used to reset GICCLK domain components
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	slave	PVBus	External debug interface.
pmbirq	master	Signal	Interrupt signal from the statistical profiling unit.
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	master	Signal	PPU cluster interrupt.
ppu_cluster_isolate	master	Signal	PPU Cluster Domain isolation control.
ppu_cluster_wakerequest	slave	Signal	PPU cluster wake request signal.
ppu_cme_irq	master	Signal	PPU CME interrupt
ppu_core_irq	master	Signal	PPU core interrupt.
ppu_core_isolate	master	Signal	PPU Core Domain isolation control.
ppu_core_wakerequest	slave	Signal	PPU core wake request signal.
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	master	PVBus	The core will generate bus requests on this port.
pvbus_periph_m	master	PVBus	The core can generate peripheral bus request on this port.
rei	slave	Signal	Per core RAM Error Interrupt.
reset	slave	Signal	Reset signal to cluster.
rvbaraddr	slave	Value_64	Reset vector base address.
sci_m	master	SystemCoherencyInterface	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei	slave	Signal	Per core System Error physical pins.
spiden	slave	Signal	External debug interface.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq	master	Signal	Interrupt signal from the trace buffer unit.
utility_bus_s	slave	PVBus	Utility bus subordinate

Port	Direction	Protocol	Description
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfiq	slave	Signal	Virtualised FIQ.
virq	slave	Signal	Virtualised IRQ.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Per core virtual System Error physical pins.

## Parameters for ARMC1UltraCT

### **cpuX.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **cpuX.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **cpuX.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **cpuX.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **cpuX.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**cpuX.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t



Default value: 0x80000

### **cpuX.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **cpuX.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **cpuX.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

### **cpuX.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **cpuX.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **cpuX.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**cpuX.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**cpuX.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**cpuX.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**cpuX.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**cpuX.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**cpuX.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**cpuX.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**cpuX.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**cpuX.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**cpuX.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**cpuX.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**cpuX.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**cpuX.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**cpuX.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`cpuX.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`cpuX.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`AEND0_DEFAULT`**

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMPO input signal).

Type: `uint64_t`

Default value: `0x0`

### **`AEND1_DEFAULT`**

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

Type: `uint64_t`

Default value: `0x0`

### **`AEND2_DEFAULT`**

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

Type: `uint64_t`

Default value: `0x0`

### **`AEND3_DEFAULT`**

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

Type: `uint64_t`

Default value: `0x0`

### **ASTART0\_DEFAULT**

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

Type: `uint64_t`

Default value: `0x0`

### **ASTART1\_DEFAULT**

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

Type: `uint64_t`

Default value: `0x0`

### **ASTART2\_DEFAULT**

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

Type: `uint64_t`

Default value: `0x0`

### **ASTART3\_DEFAULT**

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

Type: `uint64_t`

Default value: `0x0`

### **BROADCASTATOMIC**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`

Default value: `true`

### **BROADCASTCACHEMAINT**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

Type: `bool`

Default value: `false`

### **BROADCASTOUTER**

Enable broadcasting of Outer Shareable transactions. The `broadcastouter` signal will override this value if used.

Type: `bool`

Default value: `false`

### **BROADCASTPERSIST**

Enable broadcasting of cache clean to the point of persistence operations. The `broadcastpersist` signal will override this value if used.

Type: `bool`

Default value: `true`

### **CLUSTER\_ID**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the ManagerID64 of the core.

Type: `uint32_t`

Default value: `0x0`

### **CMO\_broadcast\_when\_cache\_state\_modelling\_disabled**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. This value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain events even when cache state modelling is disabled. Possible values are:- 0 = All CMOs are broadcast if architecturally required- 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required.

Type: `uint8_t`

Default value: `1`

### **CPUCFR**

Value of CPU Configuration Register.

Type: `uint64_t`

Default value: 0x1

#### **DSU.cme.CMECFR**

Value of CMECFR\_EL1 fields (ECC, CHI) for all CMEs.

Type: `string`

Default value: `{"ECC": 0, "CHI": 0}`

#### **DSU.cme.mpmmm\_accumulator\_multiplier**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator. value of `n` means the accumulator will use (`n * accumulator value`) to calculate the mpmm threshold (MPMM). is provided as a fast model workaround to handle cases where execution of quantum in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

Type: `uint8_t`

Default value: 1

#### **DSU.cme.power\_on\_by\_default**

If true, CME PPU's will initialize in Dynamic mode out of reset, enabling the CME to power itself on/off automatically as it is used.

Type: `bool`

Default value: `true`

#### **GICDISABLE**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`

Default value: `true`

#### **NUM\_CMES**

Defines how many CMEs are associated with each cluster. This parameter must be in sync with SVE.ScalableVectorExtension SME configuration.

Type: `uint8_t`

Default value: 0

#### **NUM\_CORES**

Number of cores per cluster.

Type: `uint8_t`

Default value: 1

### **bus\_type**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

Type: `uint8_t`

Default value: 0

### **core\_power\_on\_by\_default**

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPU which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

Type: `bool`

Default value: `false`

### **cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: 0x1

### **cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: 0x1

### **dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **dcache-maintenance\_latency**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0



**dcache-miss\_latency**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**dcache-prefetch\_enabled**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

Type: `bool`

Default value: `false`

**dcache-read\_access\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**dcache-size**

L1 D-Cache size in bytes.

Type: `uint32_t`

Default value: `0x10000`

**dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: 0x0

### **dcache-state\_modelled**

Set whether D-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **dcache-write\_access\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **default\_opmode**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

Type: `uint8_t`

Default value: 4

### **diagnostics**

Enable DynamIQ diagnostic messages.

Type: `bool`

Default value: `false`

### **ecv\_support\_level**

Implement Enhanced Counter Virtualization feature from ARMv8.6. 0, Not supported. 1, fully supported without CNTPOFF. 2, fully supported with CNTPOFF (FEAT\_ECV).

Type: `uint8_t`

Default value: 2

### **enable\_simulation\_performance\_optimizations**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

Type: `bool`

Default value: `true`

### **ete.CLAIMTAGS**

Number of claim tags.

Type: `uint8_t`

Default value: 4

### **ete.ETE\_REVISION**

ETE revision: 0=ETEv1.0, 1=ETEv1.1, 2=ETEv1.2, 3=ETEv1.3.

Type: `uint8_t`

Default value: 1

### **ete.MAX\_INST\_PER\_Q**

Maximum limit for the number of instructions implied by a Q element.

Type: `uint16_t`

Default value: `0x1`

### **ete.NumberOfRSPairs**

Number of resource selector pairs.

Type: `uint8_t`

Default value: 8

### **ete.PIDR\_CM0D**

TRCPIDR CM0D value.

Type: `uint8_t`

Default value: 0

**ete.Q\_CADENCE**

Number of instruction blocks traced between two Q elements.

Type: `uint16_t`

Default value: `0x1`

**ete.RES0\_STATEFUL**

Whether **RES0** bits are stateful or **RAZ/WI**.

Type: `bool`

Default value: `false`

**ete.RETSTACK**

Return stack depth.

Type: `uint8_t`

Default value: `3`

**ete.SIM\_OVERFLOW\_GRANULARITY**

Number of instruction blocks in each granule, for simulated overflow.

Type: `uint32_t`

Default value: `0x64`

**ete.SIM\_OVERFLOW\_PERCENTAGE**

Percentage of instruction blocks lost in each granule, for simulated overflow.

Type: `uint8_t`

Default value: `0`

**ete.SOURCE\_ADDRESS**

Allow generation of source address elements.

Type: `bool`

Default value: `false`

**ete.TRACE\_OUTPUT**

File to which to write trace byte stream.

Type: `string`

Default value: `N/A`

**ete.TRCRSRTA\_FORCED\_EXCEP**

TRCRSR.TA value for a forcibly traced exception.

Type: `bool`

Default value: `false`

**ext\_abort\_so\_write\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `uint8_t`

Default value: 2

**force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks**

Force MTE tag accesses to **RAZ/WI** and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

Type: `bool`

Default value: `false`

**force\_zero\_PSTATE\_PAN**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

**force\_zero\_mpam\_partid\_and\_pmg**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAMO\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

Type: `uint8_t`

Default value: 0

**has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

**has\_enhanced\_pan**

Implements Armv8.7 Enhanced PAN feature (FEAT\_PAN3) values of this parameter are:- 1, feature is implemented if ARMv8.7 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 2

**has\_ete**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (-plugin or -P).

Type: `bool`

Default value: `false`

**has\_mt\_pmu\_disable\_feature**

Implements Multi-threading PMU disable extension from ARMv8.6 (FEAT\_MTPMU). 0: FEAT\_MTPMU is disabled, 1: FEAT\_MTPMU is enabled if ARMv8.6 is implemented, 2: FEAT\_MTPMU is cherry-picked, 0xF: The feature is disabled and is represented by value 0xF in ID\_AA64DFR0\_EL1.MTPMU.

Type: `uint8_t`

Default value: 0

**has\_peripheral\_port**

If true, an additional AXI peripheral port is configured.

Type: `bool`

Default value: `false`

**has\_statistical\_profiling**

Whether Statistical Based Profiling is implemented (FEAT\_SPE).

Type: `bool`

Default value: `true`

**has\_v8\_7\_spe\_inverted\_filtering**

Where FEAT\_SPEv1p2 is implemented, whether inverted filtering by events is implemented (represented by PMISDR.FnE).

Type: `bool`

Default value: `false`

**has\_v8\_7\_spe\_previous\_branch\_target**

Where FEAT\_SPEv1p2 is implemented, whether the optional branch target feature is implemented (FEAT\_SPE\_PBT).

Type: `bool`

Default value: `false`

**icache-hit\_latency**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

**icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-size**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: `0x10000`

**icache-state\_modelled**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

**instruction\_tlb\_size**

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

Type: `uint32_t`

Default value: `0x0`

**invalidate\_code\_cache\_on\_icache\_cmo**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: `uint8_t`



Default value: 0

### **l3cache-mpamf.max\_partid\_ns**

Maximum value of non-secure PARTID supported for DSU L3Cache. Options are 7 or 63.

Type: `uint8_t`

Default value: 63

### **l3cache-size**

L3 Cache size in bytes.

Type: `uint32_t`

Default value: 0x100000

### **log2\_trace\_buffer\_alignment**

Log2 of trace buffer alignment constraint for output buffer (0->1B ... 11->2Kib).

Type: `uint8_t`

Default value: 6

### **memory\_tagging\_support\_level**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

Type: `uint8_t`

Default value: 3

### **mpam\_has\_altsp**

DEPRECATED: MPAMIDR\_EL1.HAS\_ALTSP is required when FEAT\_RME is implemented.

Type: `bool`

Default value: `false`

### **mpamidr\_has\_force\_ns**

Whether MPAMIDR\_EL1.HAS\_FORCE\_NS bit is set or clear.values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.6 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

**mpamidr\_has\_sdeflt**

Whether MPAMIDR\_EL1.HAS\_SDEFLT bit is set or clear.values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.6 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

**mpamidr\_has\_tidr**

Whether MPAMIDR\_EL1.HAS\_TIDR bit is set or clearvalues of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.6 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

**mpmm\_accumulator\_multiplier**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator.value of n means the accumulator will use (n \* accumulator value) to calculate the mpmm threshold (MPMM).is provided as a fast model workaround to handle cases where execution of quantum in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

Type: uint8\_t

Default value: 1

**num\_acp**

Number of ACP ports.

Type: uint8\_t

Default value: 0

**num\_nodes**

Number of transport nodes. Zero implies direct-connect configuration.

Type: uint8\_t

Default value: 1

**pmu-num\_counters**

Number of PMU counters implemented.

Type: uint8\_t

Default value: 31

**ptw\_latency**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

**stage12\_tlb\_size**

Number of stage1+2 tlb entries. Valid values are 0 or  $\geq 4$ .

Type: `uint32_t`

Default value: `0x80`

**tcr\_txsz\_undersize\_should\_fault**

If large VA is not supported, Whether undersized TxSZ value should generate translation fault.

Type: `bool`

Default value: `false`

**tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

**tlbi\_stall\_enabled**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: `bool`

Default value: `false`

**treat\_PAC\_as\_NOP**

Non-architecture parameter. Treat Pointer Authentication as **NOP**. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are **NOP**. Branch, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

Type: `bool`

Default value: `false`

**walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

## 3.40 ARMCortexA320CT

Defined in `LISA/ARMCortexA320CT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

Model quality level changes:

From	To
Preliminary support	Full support

### About ARMCortexA320CT

The number of cores currently supported is 1,2,4. The following example platforms are available:

- FVP\_Base\_Cortex-A320
- EVS\_Dhrystone\_Cortex-A320x1
- SVP\_Base\_Cortex-A320x1

### Limitations

The following features are not yet supported, and will be added in a future release:

- No support for Core-Complex yet.
- BROADCASTCACHEMAINTPOU pin is not implemented.
- COREINSTRRET signals are not implemented.
- MEM\_RET power mode is not supported.

The following features are not supported in this or future releases:

- DynamIQ features that are negligible to the programmers view simulation.
- 256-bit wide output transactions.
- Error correction/detection features.

- Self-test features (MBIST).
- Latency configuration.
- Snoop filtering.
- Cache stashing capability.

## Iris and MTI instances for ARM Cortex-A320CT

This model has the following Iris instances:

Name	Instance type
ARM Cortex-A320CT	Cluster_ARM_Cortex-A320
ARM Cortex-A320CT.AMU	PVBusLogger
ARM Cortex-A320CT.AMU.mapper	PVBusMapper
ARM Cortex-A320CT.DAP	PVBusLogger
ARM Cortex-A320CT.DAP.mapper	PVBusMapper
ARM Cortex-A320CT.DSU	DSU-120
ARM Cortex-A320CT.DSU.PPU_cluster	PPUv1
ARM Cortex-A320CT.DSU.PPU_cluster.busslave	PVBusSlave
ARM Cortex-A320CT.DSU.PPU_core0	PPUv1
ARM Cortex-A320CT.DSU.PPU_core0.busslave	PVBusSlave
ARM Cortex-A320CT.DSU.mpam_busslave	PVBusSlave
ARM Cortex-A320CT.DSU.shared_cache	PVCache
ARM Cortex-A320CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARM Cortex-A320CT.DSU.shared_cache.upstream[Y] (where Y = 0-4)	PVBusSlave
ARM Cortex-A320CT.DSU.utility_slave[0]	PVBusSlave
ARM Cortex-A320CT.MMAP	PVBusLogger
ARM Cortex-A320CT.MMAP.mapper	PVBusMapper
ARM Cortex-A320CT.RAS	PVBusLogger
ARM Cortex-A320CT.RAS.mapper	PVBusMapper
ARM Cortex-A320CT.cpu0	ARM_Cortex-A320
ARM Cortex-A320CT.cpu0.UTLB	TLB
ARM Cortex-A320CT.cpu0.debug_rom	debug_rom
ARM Cortex-A320CT.cpu0.dtlb	TLB
ARM Cortex-A320CT.cpu0.l1dcache	PVCache
ARM Cortex-A320CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARM Cortex-A320CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARM Cortex-A320CT.cpu0.l1licache	PVCache
ARM Cortex-A320CT.cpu0.l1licache.downstream[0].pvbusmaster	PVBusMaster
ARM Cortex-A320CT.cpu0.l1licache.upstream[0]	PVBusSlave
ARM Cortex-A320CT.ext_bus	PVBusLogger
ARM Cortex-A320CT.ext_bus.mapper	PVBusMapper
ARM Cortex-A320CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

Name	Instance type
ARMCortexA320CT.global_debug_rom	debug_rom
ARMCortexA320CT.l2_cache	PVCache
ARMCortexA320CT.l2_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA320CT.l2_cache.upstream[Z] (where Z = 0-16)	PVBusSlave
ARMCortexA320CT.secondary_debug_rom	debug_rom
ARMCortexA320CT.sve	ScalableVectorExtension

This model has the following MTI trace components:

Name	Component type
ARMCortexA320CT.AMU	PVBusLogger
ARMCortexA320CT.AMU.mapper	PVBusMapper
ARMCortexA320CT.DAP	PVBusLogger
ARMCortexA320CT.DAP.mapper	PVBusMapper
ARMCortexA320CT.DSU	DSU-120
ARMCortexA320CT.DSU.PPU_cluster	PPUv1
ARMCortexA320CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA320CT.DSU.PPU_core0	PPUv1
ARMCortexA320CT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexA320CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA320CT.DSU.shared_cache	PVCache
ARMCortexA320CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA320CT.DSU.shared_cache.upstream[Y] (where Y = 0-4)	PVBusSlave
ARMCortexA320CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA320CT.MMAP	PVBusLogger
ARMCortexA320CT.MMAP.mapper	PVBusMapper
ARMCortexA320CT.RAS	PVBusLogger
ARMCortexA320CT.RAS.mapper	PVBusMapper
ARMCortexA320CT.cpu0	ARM_Cortex-A320
ARMCortexA320CT.cpu0.UTLB	TLB
ARMCortexA320CT.cpu0.l1dcache	PVCache
ARMCortexA320CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA320CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA320CT.cpu0.l1icache	PVCache
ARMCortexA320CT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA320CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA320CT.ext_bus	PVBusLogger
ARMCortexA320CT.ext_bus.mapper	PVBusMapper
ARMCortexA320CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA320CT.l2_cache	PVCache
ARMCortexA320CT.l2_cache.downstream[0].pvbusmaster	PVBusMaster

Name	Component type
ARMCortexA320CT.12_cache.upstream[Z] (where Z = 0-16)	PVBusSlave

## Ports for ARMCortexA320CT

Port	Direction	Protocol	Description
acp_s	slave	PVBus	AXI ACP subordinate port.
AEND0MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND2MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND3MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART2MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	slave	Signal	CHI defined pins.
broadcastcachemaint	slave	Signal	ACE defined pins.
broadcastouter	slave	Signal	ACE defined pins.
broadcastpersist	slave	Signal	CHI defined pins.
cfgend	slave	Signal	This signal is for EE bit initialisation.
cfgsdisable	slave	Signal	This signal disables write access to some secure Interrupt Controller registers.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcs_m_pchannel	master	PChannel	Cluster PCSM signal
cluster_powerdown_out	master	Signal	DEPRECATED - An output from PPUs that informs thermal controller of the cluster power info
cluster_ppu_hw_stat	master	Value	Cluster PPU power state
clusterid	slave	Value	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set ManagerID64 of the core.

Port	Direction	Protocol	Description
clusterpmuirq	master	Signal	DynamlQ pmu irq
CNTHPIRQ	master	Signal	Timer signals to SOC.
CNTHPSIRQ	master	Signal	Timer signals to SOC
CNTHVIRQ	master	Signal	Timer signals to SOC.
CNTHVSIRQ	master	Signal	Timer signals to SOC
CNTPNSIRQ	master	Signal	Timer signals to SOC.
CNTPSIRQ	master	Signal	Timer signals to SOC.
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.
CNTVIRQ	master	Signal	Timer signals to SOC.
commirq	master	Signal	Interrupt signal from debug communications channel.
complex_pcs_m_pchannel	master	PChannel	Complex PCSM signals
complexerrirq	master	Signal	Error indicator for an ECC error that causes potential data corruption or loss of coherency
complexfaultirq	master	Signal	Fault indicator for a detected 1 or 2 bit ECC error
core_clk_in	slave	ClockSignal	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
core_pcs_m_pchannel	master	PChannel	Core PCSM signals
core_powerdown_out	master	Signal	DEPRECATED - An output from PPU that informs thermal controller of the core power info
core_ppu_hw_stat	master	Value	Notify the power state change inside the PPU.
coreerrirq	master	Signal	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq	master	Signal	Fault indicator for a detected 1 or 2 bit ECC error
coreinstrrun	master	Signal	Core Running State
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
ctidbgirq	master	Signal	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	slave	Signal	External debug interface.
dbgnopwrdown	master	Signal	No power-down request.
dbgpwrupreq	master	Signal	Debug power up request.
defaultMP	slave	Signal	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main manager interface ports.
dev_debug_s	slave	PVBus	External debug interface.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.
gicreset	master	Signal	An output from PPU that can be used to reset GICCLK domain components
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.



Port	Direction	Protocol	Description
irq	slave	Signal	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	slave	PVBus	External debug interface.
pmbirq	master	Signal	Interrupt signal from the statistical profiling unit.
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	master	Signal	PPU cluster interrupt.
ppu_cluster_isolate	master	Signal	PPU Cluster Domain isolation control.
ppu_cluster_wakerequest	slave	Signal	PPU cluster wake request signal.
ppu_core_irq	master	Signal	PPU core interrupt.
ppu_core_isolate	master	Signal	PPU Core Domain isolation control.
ppu_core_wakerequest	slave	Signal	PPU core wake request signal.
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	master	PVBus	The core will generate bus requests on this port.
pvbus_periph_m	master	PVBus	The core can generate peripheral bus request on this port.
rei	slave	Signal	Per core RAM Error Interrupt.
reset	slave	Signal	Reset signal to cluster.
rvbaraddr	slave	Value_64	Reset vector base address.
sci_m	master	SystemCoherencyInterface	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei	slave	Signal	Per core System Error physical pins.
spiden	slave	Signal	External debug interface.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq	master	Signal	Interrupt signal from the trace buffer unit.
utility_bus_s	slave	PVBus	Utility bus subordinate
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfiq	slave	Signal	Virtualised FIQ.
virq	slave	Signal	Virtualised IRQ.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Per core virtual System Error physical pins.

## Parameters for ARM Cortex A320CT

### cpuX.CFGEND

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: `false`

**cpuX.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**cpuX.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**cpuX.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**cpuX.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

**cpuX.min\_sync\_level**

Force minimum `syncLevel` (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

**cpuX.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**cpuX.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`cpuX.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

### **`cpuX.semihosting-T32_HLT`**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

### **`cpuX.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

### **`cpuX.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

### **`cpuX.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

### **`cpuX.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`cpuX.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`cpuX.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`cpuX.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`cpuX.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`cpuX.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`cpuX.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**AEND0\_DEFAULT**

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

Type: `uint64_t`

Default value: `0x0`

**AEND1\_DEFAULT**

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

Type: `uint64_t`

Default value: `0x0`

**AEND2\_DEFAULT**

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

Type: `uint64_t`

Default value: `0x0`

**AEND3\_DEFAULT**

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART0\_DEFAULT**

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART1\_DEFAULT**

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART2\_DEFAULT**

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART3\_DEFAULT**

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

Type: `uint64_t`

Default value: `0x0`

**BROADCASTATOMIC**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`

Default value: `true`

**BROADCASTCACHEMAINT**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

Type: `bool`

Default value: `false`

**BROADCASTOUTER**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

Type: `bool`

Default value: `false`

**BROADCASTPERSIST**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: `bool`

Default value: `true`

**CLUSTER\_ID**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the ManagerID64 of the core.

Type: `uint32_t`

Default value: `0x0`

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. This value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required.

Type: `uint8_t`

Default value: `1`

**CPUCFR**

Value of CPU Configuration Register.

Type: `uint64_t`

Default value: `0x80000`

**DBGROMADDR**

Initialization value of DBGDRAR register. Bits[55:12] of this register specify the ROM table physical address.

Type: `uint64_t`

Default value: `0x0`

**DBGROMADDRV**

If true, set bits[1:0] of the CP15 DBGDRAR to indicate that the address is valid.

Type: `bool`

Default value: `false`

**GICDISABLE**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`

Default value: `true`

**NUM\_CORES**

Number of cores per cluster.

Type: `uint8_t`

Default value: `1`

**bus\_type**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

Type: `uint8_t`

Default value: `0`

**core\_cache\_protection**

`core_cache_protection` can change `ERR0FR`, `ERR0PFGF` and `ERR0PFGCTL` fields. Possible values are: -1:Not implemented (by default), 0:Disabled, 1:Enabled.

Type: `int8_t`

Default value: `1`

**core\_power\_on\_by\_default**

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPUs which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

Type: `bool`

Default value: `false`

**cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`



**cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-maintenance\_latency**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-miss\_latency**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-prefetch\_enabled**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

Type: `bool`

Default value: `false`

**dcache-read\_access\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-size**

L1 D-Cache size in bytes.

Type: `uint32_t`

Default value: `0x8000`

**dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-state\_modelled**

Set whether D-cache has stateful implementation.

Type: `bool`

Default value: `false`

**dcache-write\_access\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **enable\_simulation\_performance\_optimizations**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

Type: bool

Default value: true

### **ete.CLAIMTAGS**

Number of claim tags.

Type: uint8\_t

Default value: 4

### **ete.MAX\_INST\_PER\_Q**

Maximum limit for the number of instructions implied by a Q element.

Type: uint16\_t

Default value: 0x1

### **ete.PIDR\_CM0D**

TRCPIDR CM0D value.

Type: uint8\_t

Default value: 0

### **ete.PIDR\_REVAND**

TRCPIDR REVAND value.

Type: uint8\_t

Default value: 0

### **ete.PIDR\_REVISION**

TRCPIDR REVISION value.

Type: uint8\_t

Default value: 0

**ete.Q\_CADENCE**

Number of instruction blocks traced between two Q elements.

Type: `uint16_t`

Default value: `0x1`

**ete.RES0\_STATEFUL**

Whether **RES0** bits are stateful or **RAZ/WI**.

Type: `bool`

Default value: `false`

**ete.RETSTACK**

Return stack depth.

Type: `uint8_t`

Default value: `3`

**ete.REVISION**

TRCIDR1 revision value.

Type: `uint8_t`

Default value: `0`

**ete.SIM\_OVERFLOW\_GRANULARITY**

Number of instruction blocks in each granule, for simulated overflow.

Type: `uint32_t`

Default value: `0x64`

**ete.SIM\_OVERFLOW\_PERCENTAGE**

Percentage of instruction blocks lost in each granule, for simulated overflow.

Type: `uint8_t`

Default value: `0`

**ete.SOURCE\_ADDRESS**

Allow generation of source address elements.

Type: `bool`

Default value: `false`

**ete.TRACE\_OUTPUT**

File to which to write trace byte stream.

Type: `string`

Default value: `N/A`

**ete.TRCRSRTA\_FORCED\_EXCEP**

TRCRSR.TA value for a forcibly traced exception.

Type: `bool`

Default value: `false`

**force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks**

Force MTE tag accesses to **RAZ/WI** and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

Type: `bool`

Default value: `false`

**force\_zero\_PSTATE\_PAN**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

**force\_zero\_mpam\_partid\_and\_pmg**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAMO\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

Type: `uint8_t`

Default value: `0`

**has\_actlr2**

If true ACTLR2 exists and ACTLR2(NS) is aliased to ACTLR\_EL1[63:32].

Type: `bool`

Default value: `true`

**has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

**has\_dot\_product**

Possible values of this parameter are:- 1, feature not implemented.- 2, feature is implemented.

Type: `uint8_t`

Default value: 2

**has\_ete**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (`-plugin` or `-P`).

Type: `bool`

Default value: `false`

**has\_peripheral\_port**

If true, an additional AXI peripheral port is configured.

Type: `bool`

Default value: `false`

**icache-hit\_latency**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

### **icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **icache-size**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: `0x8000`

**icache-state\_modelled**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

**invalidate\_code\_cache\_on\_icache\_cmo**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: `uint8_t`

Default value: `0`

**l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`



**l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

**l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **memory\_tagging\_support\_level**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

Type: `uint8_t`

Default value: 3

### **mpmm\_accumulator\_multiplier**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator.value of n means the accumulator will use (n \* accumulator value) to calculate the mpmm threshold (MPMM).is provided as a fast model workaround to handle cases where execution of quantum in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

Type: `uint8_t`

Default value: 1

### **pmu-num\_counters**

Number of PMU counters implemented.

Type: `uint8_t`

Default value: 6

### **ptw\_latency**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

### **ras\_extra\_configurations**

Miscellaneous configurations for error records. An array of JSON objects. Note for ERXCTLR\_EL1 register it only allows to define the mask value for the IMPDEF fields, ie bits [63:32] and bit 1, but its reset value applies on all fields. Note for ERXMISCN masks - these are 64 bit masks covering the 64 bit registers ERXMISCN\_EL1. E.g. [{"Index": 0, "ERXMISCO\_mask": 0x0, "ERXMISCO\_reset": 0x0, "ERXMISC1\_mask": 0x0, "ERXMISC1\_reset": 0x0, "ERXMISC2\_mask": 0x0, "ERXMISC2\_reset": 0x0, "ERXMISC3\_mask": 0x0, "ERXMISC3\_reset": 0x0, "ERXCTLR\_EL1\_mask": 0x0, "ERXCTLR\_EL1\_reset": 0x0}, {"Index": 1, "ERXMISCO\_mask": 0x0, "ERXMISCO\_reset": 0x0, "ERXSTATUS\_IERR\_mask": 0x300}].

Type: `string`

Default value: "[ { \"Index\": 0, \"ERXMISC0\_mask\": 0xFFFFFC0003FCF, \"ERXMISC1\_mask\": 0x03F87000FFF30F07, \"ERXPGCTL\_reset\": 0x1000}, { \"Index\": 1, \"ERXMISC0\_mask\": 0xFFFFE007FFCF, \"ERXMISC0\_reset\": 0x2, \"ERXSTATUS\_IERR\_mask\": 0x300, \"ERXMISC1\_mask\": 0x0FF8700FFFF31F0F, \"ERXPGCTL\_reset\": 0x1000} ]"

### **ras\_pfg\_clock\_mhz**

RAS Pseudo-Fault generation clock rate in MHz.

Type: `uint8_t`

Default value: 12

### **revision\_number**

Revision number of TRM implemented by the model. Changing the value can change the model behaviour. Visible as the Variant field in MIDR/MIDR\_EL1. Corresponds to the revision number X in rXpY.

Type: `uint8_t`

Default value: 0

### **store\_excl\_fail\_tag\_check\_action**

Behavior of tag check by core when a store exclusive fails. 0, Tag check ignored, 1, Tag check done if exclusive fails by global monitor.

Type: `uint8_t`

Default value: 0

### **tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

### **tlbi\_stall\_enabled**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: `bool`

Default value: `false`

### **treat\_PAC\_as\_NOP**

Non-architecture parameter. Treat Pointer Authentication as **NOP**. When the parameter is true behaviour of FEAT\_PAAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions

are **NOP**. Branch, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAAuth traps.

Type: `bool`

Default value: `false`

### **walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

## 3.41 ARMCortexA32CT

Defined in `LISA/ARMCortexA32CT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following parameters were added:

- `has_delayed_dbgreg`
- `has_delayed_sysreg`

### About ARMCortexA32CT

- If either L1 cache is stateful, then the L2 cache is stateful. This is controlled by the `dcache-state_modelled` and `icache-state_modelled` parameters.
- The cache latency parameters are only effective when you enable cache-state modeling.
- Timing annotation for transactions downstream of the cache and TLB models propagates through the models.
- Although Neon support is optional for the Arm(R) Cortex(R)-A32 processor, this model does not implement the `ase-present` parameter. Therefore, it is not possible to configure the model to not support Neon.
- This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (`crypto.dll` or `crypto.so`) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

- The `semihosting-cwd` parameter sets the current working directory that is used for semihosting. The host operating system limits the maximum path length. The `semihosting-cwd` parameter does not provide any security. Software running on the model can access files outside this directory using relative paths containing `..` or using absolute paths.
- The system designer decides whether a debug APB ties the external debug view with other system views. In the model, use a `PVBusDecoder` to direct traffic to the correct port, `dev_debug_s` or `memorymapped_debug_s`.
- The `vfp-enable_at_reset` option is model-specific and has no hardware equivalent. Arm recommends that it is only used in test systems and tied off to false in production systems.

This model has a variable number of cores per cluster, specified using the `NUM_CORES` parameter. - ECC and parity schemes are hardware-specific so are not supported.

## Iris and MTI instances for ARM CortexA32CT

This model has the following Iris instances:

Name	Instance type
ARMCortexA32CT	Cluster_ARM_Cortex-A32
ARMCortexA32CT.AMU	PVBusLogger
ARMCortexA32CT.AMU.mapper	PVBusMapper
ARMCortexA32CT.DAP	PVBusLogger
ARMCortexA32CT.DAP.mapper	PVBusMapper
ARMCortexA32CT.DSU	DSU
ARMCortexA32CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA32CT.MMAP	PVBusLogger
ARMCortexA32CT.MMAP.mapper	PVBusMapper
ARMCortexA32CT.RAS	PVBusLogger
ARMCortexA32CT.RAS.mapper	PVBusMapper
ARMCortexA32CT.acp_mapper	PVBusMapper
ARMCortexA32CT.cpu0	ARM_Cortex-A32
ARMCortexA32CT.cpu0.SZTLB (where Z = 1-2)	TLB
ARMCortexA32CT.cpu0.UTLB	TLB
ARMCortexA32CT.cpu0.dtlb	TLB
ARMCortexA32CT.cpu0.l1dcache	PVCache
ARMCortexA32CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA32CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA32CT.cpu0.l1icache	PVCache
ARMCortexA32CT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA32CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA32CT.ext_bus	PVBusLogger
ARMCortexA32CT.ext_bus.mapper	PVBusMapper
ARMCortexA32CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA32CT.global_debug_rom	debug_rom

Name	Instance type
ARMCortexA32CT.l2_cache	PVCache
ARMCortexA32CT.l2_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA32CT.l2_cache.upstream[Z] (where Z = 0-16)	PVBusSlave
ARMCortexA32CT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

Name	Component type
ARMCortexA32CT.AMU	PVBusLogger
ARMCortexA32CT.AMU.mapper	PVBusMapper
ARMCortexA32CT.DAP	PVBusLogger
ARMCortexA32CT.DAP.mapper	PVBusMapper
ARMCortexA32CT.DSU	DSU
ARMCortexA32CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA32CT.MMAP	PVBusLogger
ARMCortexA32CT.MMAP.mapper	PVBusMapper
ARMCortexA32CT.RAS	PVBusLogger
ARMCortexA32CT.RAS.mapper	PVBusMapper
ARMCortexA32CT.acp_mapper	PVBusMapper
ARMCortexA32CT.cpu0	ARM_Cortex-A32
ARMCortexA32CT.cpu0.SZTLB (where Z = 1-2)	TLB
ARMCortexA32CT.cpu0.UTLB	TLB
ARMCortexA32CT.cpu0.l1dcache	PVCache
ARMCortexA32CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA32CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA32CT.cpu0.l1icache	PVCache
ARMCortexA32CT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA32CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA32CT.ext_bus	PVBusLogger
ARMCortexA32CT.ext_bus.mapper	PVBusMapper
ARMCortexA32CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA32CT.l2_cache	PVCache
ARMCortexA32CT.l2_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA32CT.l2_cache.upstream[Z] (where Z = 0-16)	PVBusSlave
ARMCortexA32CT.l2_flusher	AsyncCacheFlushUnit

## Ports for ARMCortexA32CT

Port	Direction	Protocol	Description
acp_s	slave	PVBus	AXI ACP slave port.
broadcastcachemaint	slave	Signal	ACE defined pins.
broadcastinner	slave	Signal	ACE defined pins.

Port	Direction	Protocol	Description
broadcastouter	slave	Signal	ACE defined pins.
cfgend	slave	Signal	This signal is for EE bit initialisation.
cfgsdisable	slave	Signal	This signal disables write access to some secure Interrupt Controller registers.
cfgte	slave	Signal	This signal provides default exception handling state.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
clrexmonack	master	Signal	Acknowledge handshake signal for the clrexmonreq signal
clrexmonreq	slave	Signal	Signals the clearing of an external global exclusive monitor
clusterid	slave	Value	The port sets the value of the affinity levels 1 and 2; bits [23:16] and [15:8] of the MPIDR.
CNTHPIRQ	master	Signal	Timer signals to SOC.
CNTPNSIRQ	master	Signal	Timer signals to SOC.
CNTPSIRQ	master	Signal	Timer signals to SOC.
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.
CNTVIRQ	master	Signal	Timer signals to SOC.
commirq	master	Signal	Interrupt signal from debug communications channel.
commrx	master	Signal	Receive portion of Data Transfer Register full.
commtx	master	Signal	Transmit portion of Data Transfer Register empty.
cp15sdisable2	slave	Signal	-
cp15sdisable	slave	Signal	This signal disables write access to some system control processor registers.
cpuporeset	slave	Signal	CPU power on reset.
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
ctidbgirq	master	Signal	Cross trigger matrix port.
dbgack	master	Signal	External debug interface.
dbgen	slave	Signal	External debug interface.
dbgnopwrdown	master	Signal	No power-down request.
dbgpwrupreq	master	Signal	Processor powerup request.
dev_debug_s	slave	PVBus	External debug interface.
edbgrq	slave	Signal	External debug interface.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
l2flushdone	master	Signal	Flush of L2 memory system complete.
l2flushreq	slave	Signal	Request flush of L2 memory system.

Port	Direction	Protocol	Description
l2reset	slave	Signal	Level2 reset.
memorymapped_debug_s	slave	PVBus	External debug interface.
niden	slave	Signal	External debug interface.
periphbase	slave	Value_64	This port sets the base address of private peripheral region.
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
presetdbg	slave	Signal	Debug reset.
pvbus_m0	master	PVBus	The core will generate bus requests on this port.
rei	slave	Signal	Per core RAM Error Interrupt
reset	slave	Signal	Reset.
romaddr	slave	Value_64	Debug ROM base address.
romaddrv	slave	Signal	Debug ROM base address valid.
sei	slave	Signal	Per core System Error physical pins.
smpen	master	Signal	This signals AMP or SMP mode for each core.
spiden	slave	Signal	External debug interface.
spniden	slave	Signal	External debug interface.
standbywfe	master	Signal	This signal indicates if a core is in WFE state.
standbywfi	master	Signal	This signal indicates if a core is in WFI state
standbywfi12	master	Signal	This signal indicated all cores and L2 are idles and in low power state.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfiq	slave	Signal	Virtualised FIQ.
vinithi	slave	Signal	This signal controls of the location of the exception vectors at reset.
virq	slave	Signal	Virtualised IRQ.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Per core virtual System Error physical pins.

## Parameters for ARM Cortex A32CT

### cpuX.CFGEND

Type: `bool`

Default value: `false`

### cpuX.CFGTE

Type: `bool`

Default value: `false`



**cpuX.CP15SDISABLE**Type: `bool`Default value: `false`**cpuX.CP15SDISABLE2**Type: `bool`Default value: `false`**cpuX.CRYPTODISABLE**Type: `bool`Default value: `false`**cpuX.VINITHI**Type: `bool`Default value: `false`**cpuX.enable\_trace\_special\_hlt\_imm16**Enable usage of parameter `trace_special_hlt_imm16`.Type: `bool`Default value: `false`**cpuX.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint64_t`

Default value: 256

**cpuX.min\_sync\_level**force minimum `syncLevel` (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).Type: `uint32_t`

Default value: 0

**cpuX.semihosting-A32\_HLT**Type: `uint32_t`

Default value: 0xF000

**cpuX.semihosting-ARM\_SVC**

Type: uint32\_t

Default value: 0x123456

**cpuX.semihosting-T32\_HLT**

Type: uint32\_t

Default value: 0x3c

**cpuX.semihosting-Thumb\_SVC**

Type: uint32\_t

Default value: 0xAB

**cpuX.semihosting-cmd\_line**

Type: string

Default value: ""

**cpuX.semihosting-cwd**

Type: string

Default value: ""

**cpuX.semihosting-enable**

Type: bool

Default value: true

**cpuX.semihosting-heap\_base**

Type: uint32\_t

Default value: 0x0

**cpuX.semihosting-heap\_limit**

Type: uint32\_t

Default value: 0x0F000000

**cpuX.semihosting-stack\_base**

Type: uint32\_t

Default value: 0x10000000

**cpuX.semihosting-stack\_limit**Type: `uint32_t`Default value: `0x0F000000`**cpuX.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`Default value: `0xf000`**cpuX.vfp-enable\_at\_reset**Type: `bool`Default value: `false`**BROADCASTCACHEMAINT**

Enable broadcasting of cache maintenance operations to downstream caches. The `broadcastcachemaint` signal will override this value if used.

Type: `bool`Default value: `true`**BROADCASTINNER**

Enable broadcasting of Inner Shareable transactions. The `broadcastinner` signal will override this value if used.

Type: `bool`Default value: `true`**BROADCASTOUTER**

Enable broadcasting of Outer Shareable transactions. The `broadcastouter` signal will override this value if used.

Type: `bool`Default value: `true`**CLUSTER\_ID**Type: `uint32_t`Default value: `0`

**DBGROMADDR**

Type: uint64\_t

Default value: 0x22000000

**DBGROMADDRV**

Type: bool

Default value: true

**GICDISABLE**

Type: bool

Default value: true

**NUM\_CORES**

Number of cores in cluster.

Type: uint8\_t

Default value: 1

**PERIPHBASE**

Type: uint64\_t

Default value: 0x13080000

**bus\_type**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

Type: uint32\_t

Default value: 0

**cpi\_div**

divider for calculating CPI (Cycles Per Instruction).

Type: uint32\_t

Default value: 1

**cpi\_mul**

multiplier for calculating CPI (Cycles Per Instruction).

Type: uint32\_t

Default value: 1

**dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit.

Type: `uint64_t`

Default value: 0

**dcache-maintenance\_latency**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

**dcache-miss\_latency**

L1 D-Cache timing annotation latency for miss.

Type: `uint64_t`

Default value: 0

**dcache-read\_access\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (based on the size of `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

**dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

**dcache-size**

L1 D-Cache size in bytes.

Type: `uint32_t`

Default value: 0x8000

**dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0`

**dcache-state\_modelled**

Type: `bool`

Default value: `false`

**dcache-write\_access\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (based on the size of `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0`

**dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0`

**has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

**icache-hit\_latency**

L1 I-Cache timing annotation latency for hit.

Type: `uint64_t`

Default value: 0

### **icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

### **icache-miss\_latency**

L1 I-Cache timing annotation latency for miss.

Type: `uint64_t`

Default value: 0

### **icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

### **icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (based on the size of `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

### **icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

### **icache-size**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: `0x8000`

### **`icache-state_modelled`**

Type: `bool`

Default value: `false`

### **`l2cache-hit_latency`**

L2 Cache timing annotation latency for hit.

Type: `uint64_t`

Default value: `0`

### **`l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0`

### **`l2cache-miss_latency`**

L2 Cache timing annotation latency for miss.

Type: `uint64_t`

Default value: `0`

### **`l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access (based on the size of `l2cache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0`

### **`l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0`



**l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

**l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0`

**l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0`

**l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access (based on the size of `l2cache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0`

**l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0`

**ptw\_latency**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint32_t`

Default value: `0`

**tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint32\_t

Default value: 0

## 3.42 ARMCortexA34CT

Defined in `LISA/ARMCortexA34CT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following parameters were added:

- `has_delayed_dbgreg`
- `has_delayed_sysreg`

### About ARMCortexA34CT

The model has the following features:

- If either L1 cache is stateful, then the L2 cache is stateful. This is controlled by the `dcache-state_modelled` and `icache-state_modelled` parameters.
- The cache latency parameters are only effective when you enable cache-state modeling.
- Timing annotation for transactions downstream of the cache and TLB models propagates through the models.
- Although Neon support is optional for this processor, this model does not implement the `ase-present` parameter. Therefore, it is not possible to configure the model to not support Neon.
- This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (`crypto.dll` or `crypto.so`) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).
- The `semihosting-cwd` parameter sets the current working directory that is used for semihosting. The host operating system limits the maximum path length. The `semihosting-cwd` parameter does not provide any security. Software running on the model can access files outside this directory using relative paths containing `..` or using absolute paths.
- The system designer decides whether a debug APB ties the external debug view with other system views. In the model, use a PVBUSDecoder to direct traffic to the correct port, `dev_debug_s` or `memorymapped_debug_s`.

- The `vfp-enable_at_reset` option is model-specific and has no hardware equivalent. Arm recommends that it is only used in test systems and tied off to false in production systems.
- This model has a variable number of cores per cluster, specified using the `NUM_CORES` parameter.
- ECC and parity schemes are hardware-specific so are not supported.

## Iris and MTI instances for ARMCortexA34CT

This model has the following Iris instances:

Name	Instance type
ARMCortexA34CT	Cluster_ARM_Cortex-A34
ARMCortexA34CT.AMU	PVBusLogger
ARMCortexA34CT.AMU.mapper	PVBusMapper
ARMCortexA34CT.DAP	PVBusLogger
ARMCortexA34CT.DAP.mapper	PVBusMapper
ARMCortexA34CT.DSU	DSU
ARMCortexA34CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA34CT.MMAP	PVBusLogger
ARMCortexA34CT.MMAP.mapper	PVBusMapper
ARMCortexA34CT.RAS	PVBusLogger
ARMCortexA34CT.RAS.mapper	PVBusMapper
ARMCortexA34CT.acp_mapper	PVBusMapper
ARMCortexA34CT.cpu0	ARM_Cortex-A34
ARMCortexA34CT.cpu0.UTLB	TLB
ARMCortexA34CT.cpu0.dtlb	TLB
ARMCortexA34CT.cpu0.l1dcache	PVCache
ARMCortexA34CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA34CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA34CT.cpu0.l1icache	PVCache
ARMCortexA34CT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA34CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA34CT.ext_bus	PVBusLogger
ARMCortexA34CT.ext_bus.mapper	PVBusMapper
ARMCortexA34CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA34CT.global_debug_rom	debug_rom
ARMCortexA34CT.l2_cache	PVCache
ARMCortexA34CT.l2_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA34CT.l2_cache.upstream[Z] (where Z = 0-16)	PVBusSlave
ARMCortexA34CT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

Name	Component type
ARMCortexA34CT.AMU	PVBusLogger
ARMCortexA34CT.AMU.mapper	PVBusMapper
ARMCortexA34CT.DAP	PVBusLogger
ARMCortexA34CT.DAP.mapper	PVBusMapper
ARMCortexA34CT.DSU	DSU
ARMCortexA34CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA34CT.MMAP	PVBusLogger
ARMCortexA34CT.MMAP.mapper	PVBusMapper
ARMCortexA34CT.RAS	PVBusLogger
ARMCortexA34CT.RAS.mapper	PVBusMapper
ARMCortexA34CT.acp_mapper	PVBusMapper
ARMCortexA34CT.cpu0	ARM_Cortex-A34
ARMCortexA34CT.cpu0.UTLB	TLB
ARMCortexA34CT.cpu0.l1dcache	PVCache
ARMCortexA34CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA34CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA34CT.cpu0.l1icache	PVCache
ARMCortexA34CT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA34CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA34CT.ext_bus	PVBusLogger
ARMCortexA34CT.ext_bus.mapper	PVBusMapper
ARMCortexA34CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA34CT.l2_cache	PVCache
ARMCortexA34CT.l2_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA34CT.l2_cache.upstream[Z] (where Z = 0-16)	PVBusSlave
ARMCortexA34CT.l2_flusher	AsyncCacheFlushUnit

## Ports for ARMCortexA34CT

Port	Direction	Protocol	Description
acp_s	slave	PVBus	AXI ACP slave port.
broadcastcachemaint	slave	Signal	ACE defined pins.
broadcastinner	slave	Signal	ACE defined pins.
broadcastouter	slave	Signal	ACE defined pins.
cfgend	slave	Signal	This signal is for EE bit initialisation.
cfgsdisable	slave	Signal	This signal disables write access to some secure Interrupt Controller registers.
cfgte	slave	Signal	This signal provides default exception handling state.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.

Port	Direction	Protocol	Description
clrexmonack	master	Signal	Acknowledge handshake signal for the clrexmonreq signal
clrexmonreq	slave	Signal	Signals the clearing of an external global exclusive monitor
clusterid	slave	Value	The port sets the value of the affinity levels 1 and 2; bits [23:16] and [15:8] of the MPIDR.
CNTHPIRQ	master	Signal	Timer signals to SOC.
CNTPNSIRQ	master	Signal	Timer signals to SOC.
CNTPSIRQ	master	Signal	Timer signals to SOC.
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.
CNTVIRQ	master	Signal	Timer signals to SOC.
commirq	master	Signal	Interrupt signal from debug communications channel.
commrq	master	Signal	Receive portion of Data Transfer Register full.
commtx	master	Signal	Transmit portion of Data Transfer Register empty.
cp15sdisable	slave	Signal	This signal disables write access to some system control processor registers.
cpuporeset	slave	Signal	Power on reset. Initializes all the processor logic, including debug logic.
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
ctidbgirq	master	Signal	Cross Trigger Interface (CTI) interrupt trigger output.
dbgack	master	Signal	External debug interface.
dbgen	slave	Signal	External debug interface.
dbgnopwrdown	master	Signal	No power-down request.
dbgpwrupreq	master	Signal	Processor powerup request.
dev_debug_s	slave	PVBus	External debug interface.
edbgrq	slave	Signal	External debug interface.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
l2flushdone	master	Signal	Flush of L2 memory system complete.
l2flushreq	slave	Signal	Request flush of L2 memory system.
l2reset	slave	Signal	This signal resets the shared L2 memory system, interrupt controller and timer logic.
memorymapped_debug_s	slave	PVBus	External debug interface.
niden	slave	Signal	External debug interface.
periphbase	slave	Value_64	This port sets the base address of private peripheral region.
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.

Port	Direction	Protocol	Description
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	master	PVBus	The core will generate bus requests on this port.
rei	slave	Signal	Per core RAM Error Interrupt.
reset	slave	Signal	Raising this signal will put the core into reset mode.
romaddr	slave	Value_64	Debug ROM base address.
romaddrv	slave	Signal	Debug ROM base address valid.
rvbaraddr	slave	Value_64	Reset vector base address.
sei	slave	Signal	Per core virtual System Error physical pins.
smpen	master	Signal	This signals AMP or SMP mode for each core.
spiden	slave	Signal	External debug interface.
spniden	slave	Signal	External debug interface.
standbywfe	master	Signal	This signal indicates if a core is in WFE state.
standbywfi	master	Signal	This signal indicates if a core is in WFI state.
standbywfil2	master	Signal	Indicate that all the individual processors and the L2 memory system are in a WFI state.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfiq	slave	Signal	Virtualised FIQ.
vinithi	slave	Signal	This signal controls of the location of the exception vectors at reset.
virq	slave	Signal	Virtualised IRQ.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Per core virtual System Error physical pins.

## Parameters for ARM CortexA34CT

### **cpuX.CFGEND**

Type: `bool`

Default value: `false`

### **cpuX.CFGTE**

Type: `bool`

Default value: `false`

### **cpuX.CP15SDISABLE**

Type: `bool`

Default value: `false`

**cpuX.CRYPTODISABLE**

Type: `bool`

Default value: `false`

**cpuX.RVBARADDR**

Type: `uint64_t`

Default value: `0`

**cpuX.VINITHI**

Type: `bool`

Default value: `false`

**cpuX.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**cpuX.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint64_t`

Default value: `256`

**cpuX.min\_sync\_level**

force minimum `syncLevel` (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint32_t`

Default value: `0`

**cpuX.semihosting-A32\_HLT**

Type: `uint32_t`

Default value: `0xF000`

**cpuX.semihosting-A64\_HLT**

Type: `uint32_t`

Default value: 0xF000

**cpuX.semihosting-ARM\_SVC**

Type: uint32\_t

Default value: 0x123456

**cpuX.semihosting-T32\_HLT**

Type: uint32\_t

Default value: 0x3c

**cpuX.semihosting-Thumb\_SVC**

Type: uint32\_t

Default value: 0xAB

**cpuX.semihosting-cmd\_line**

Type: string

Default value: ""

**cpuX.semihosting-cwd**

Type: string

Default value: ""

**cpuX.semihosting-enable**

Type: bool

Default value: true

**cpuX.semihosting-heap\_base**

Type: uint32\_t

Default value: 0x0

**cpuX.semihosting-heap\_limit**

Type: uint32\_t

Default value: 0x0F000000

**cpuX.semihosting-stack\_base**

Type: uint32\_t

Default value: 0x10000000



**cpuX.semihosting-stack\_limit**Type: `uint32_t`Default value: `0x0F000000`**cpuX.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`Default value: `0xf000`**cpuX.vfp-enable\_at\_reset**Type: `bool`Default value: `false`**BROADCASTCACHEMAINT**

Enable broadcasting of cache maintenance operations to downstream caches. The `broadcastcachemaint` signal will override this value if used.

Type: `bool`Default value: `true`**BROADCASTINNER**

Enable broadcasting of Inner Shareable transactions. The `broadcastinner` signal will override this value if used.

Type: `bool`Default value: `true`**BROADCASTOUTER**

Enable broadcasting of Outer Shareable transactions. The `broadcastouter` signal will override this value if used.

Type: `bool`Default value: `true`**CLUSTER\_ID**Type: `uint32_t`Default value: `0`

**DBGROMADDR**

Type: uint64\_t

Default value: 0x22000000

**DBGROMADDRV**

Type: bool

Default value: true

**GICDISABLE**

Type: bool

Default value: true

**NUM\_CORES**

Number of cores in cluster.

Type: uint8\_t

Default value: 1

**PERIPHBASE**

Type: uint64\_t

Default value: 0x13080000

**bus\_type**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

Type: uint32\_t

Default value: 0

**cpi\_div**

divider for calculating CPI (Cycles Per Instruction).

Type: uint32\_t

Default value: 1

**cpi\_mul**

multiplier for calculating CPI (Cycles Per Instruction).

Type: uint32\_t

Default value: 1

**dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit.

Type: `uint64_t`

Default value: 0

**dcache-maintenance\_latency**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

**dcache-miss\_latency**

L1 D-Cache timing annotation latency for miss.

Type: `uint64_t`

Default value: 0

**dcache-read\_access\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (based on the size of `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

**dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

**dcache-size**

L1 D-Cache size in bytes.

Type: `uint32_t`

Default value: 0x8000

**dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0`

**dcache-state\_modelled**

Type: `bool`

Default value: `false`

**dcache-write\_access\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (based on the size of `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0`

**dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0`

**has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

**icache-hit\_latency**

L1 I-Cache timing annotation latency for hit.

Type: `uint64_t`

Default value: 0

### **icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

### **icache-miss\_latency**

L1 I-Cache timing annotation latency for miss.

Type: `uint64_t`

Default value: 0

### **icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

### **icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (based on the size of `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

### **icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

### **icache-size**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: `0x8000`

### **`icache-state_modelled`**

Type: `bool`

Default value: `false`

### **`l2cache-hit_latency`**

L2 Cache timing annotation latency for hit.

Type: `uint64_t`

Default value: `0`

### **`l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0`

### **`l2cache-miss_latency`**

L2 Cache timing annotation latency for miss.

Type: `uint64_t`

Default value: `0`

### **`l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access (based on the size of `l2cache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0`

### **`l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0`

**l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

**l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0`

**l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0`

**l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access (based on the size of `l2cache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0`

**l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0`

**ptw\_latency**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint32_t`

Default value: `0`

**tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint32\_t

Default value: 0

## 3.43 ARMCortexA35CT

Defined in `LISA/ARMCortexA35CT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following parameters were added:

- `has_delayed_dbgreg`
- `has_delayed_sysreg`

### About ARMCortexA35CT

The model has the following features:

- If either L1 cache is stateful, then the L2 cache is stateful. This is controlled by the `dcache-state_modelled` and `icache-state_modelled` parameters.
- The cache latency parameters are only effective when you enable cache-state modeling.
- Timing annotation for transactions downstream of the cache and TLB models propagates through the models.
- Although Neon support is optional for this processor, this model does not implement the `ase-present` parameter. Therefore, it is not possible to configure the model to not support Neon.
- This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (`crypto.dll` or `crypto.so`) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).
- The `semihosting-cwd` parameter sets the current working directory that is used for semihosting. The host operating system limits the maximum path length. The `semihosting-cwd` parameter does not provide any security. Software running on the model can access files outside this directory using relative paths containing `..` or using absolute paths.
- The system designer decides whether a debug APB ties the external debug view with other system views. In the model, use a PVBUSDecoder to direct traffic to the correct port, `dev_debug_s` or `memorymapped_debug_s`.



- The `vfp-enable_at_reset` option is model-specific and has no hardware equivalent. Arm recommends that it is only used in test systems and tied off to false in production systems. This model has a variable number of cores per cluster, specified using the `NUM_CORES` parameter.
- ECC and parity schemes are hardware-specific so are not supported.

### Iris and MTI instances for ARM Cortex A35CT

This model has the following Iris instances:

Name	Instance type
ARM Cortex A35CT	Cluster_ARM_Cortex-A35
ARM Cortex A35CT.AMU	PVBusLogger
ARM Cortex A35CT.AMU.mapper	PVBusMapper
ARM Cortex A35CT.DAP	PVBusLogger
ARM Cortex A35CT.DAP.mapper	PVBusMapper
ARM Cortex A35CT.DSU	DSU
ARM Cortex A35CT.DSU.mpam_busslave	PVBusSlave
ARM Cortex A35CT.MMAP	PVBusLogger
ARM Cortex A35CT.MMAP.mapper	PVBusMapper
ARM Cortex A35CT.RAS	PVBusLogger
ARM Cortex A35CT.RAS.mapper	PVBusMapper
ARM Cortex A35CT.acp_mapper	PVBusMapper
ARM Cortex A35CT.cpu0	ARM_Cortex-A35
ARM Cortex A35CT.cpu0.UTLB	TLB
ARM Cortex A35CT.cpu0.dtlb	TLB
ARM Cortex A35CT.cpu0.l1dcache	PVCache
ARM Cortex A35CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARM Cortex A35CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARM Cortex A35CT.cpu0.l1icache	PVCache
ARM Cortex A35CT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARM Cortex A35CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARM Cortex A35CT.ext_bus	PVBusLogger
ARM Cortex A35CT.ext_bus.mapper	PVBusMapper
ARM Cortex A35CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARM Cortex A35CT.global_debug_rom	debug_rom
ARM Cortex A35CT.l2_cache	PVCache
ARM Cortex A35CT.l2_cache.downstream[0].pvbusmaster	PVBusMaster
ARM Cortex A35CT.l2_cache.upstream[Z] (where Z = 0-16)	PVBusSlave
ARM Cortex A35CT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

Name	Component type
ARM Cortex A35CT.AMU	PVBusLogger

Name	Component type
ARMCortexA35CT.AMU.mapper	PVBusMapper
ARMCortexA35CT.DAP	PVBusLogger
ARMCortexA35CT.DAP.mapper	PVBusMapper
ARMCortexA35CT.DSU	DSU
ARMCortexA35CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA35CT.MMAP	PVBusLogger
ARMCortexA35CT.MMAP.mapper	PVBusMapper
ARMCortexA35CT.RAS	PVBusLogger
ARMCortexA35CT.RAS.mapper	PVBusMapper
ARMCortexA35CT.acp_mapper	PVBusMapper
ARMCortexA35CT.cpu0	ARM_Cortex-A35
ARMCortexA35CT.cpu0.UTLB	TLB
ARMCortexA35CT.cpu0.l1dcache	PVCache
ARMCortexA35CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA35CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA35CT.cpu0.l1icache	PVCache
ARMCortexA35CT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA35CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA35CT.ext_bus	PVBusLogger
ARMCortexA35CT.ext_bus.mapper	PVBusMapper
ARMCortexA35CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA35CT.l2_cache	PVCache
ARMCortexA35CT.l2_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA35CT.l2_cache.upstream[Z] (where Z = 0-16)	PVBusSlave
ARMCortexA35CT.l2_flusher	AsyncCacheFlushUnit

## Ports for ARMCortexA35CT

Port	Direction	Protocol	Description
aa64naa32	slave	Signal	Register width after reset.
acp_s	slave	PVBus	AXI ACP slave port.
broadcastcachemaint	slave	Signal	ACE defined pins.
broadcastinner	slave	Signal	ACE defined pins.
broadcastouter	slave	Signal	ACE defined pins.
cfgend	slave	Signal	This signal is for EE bit initialisation.
cfgsdisable	slave	Signal	This signal disables write access to some secure Interrupt Controller registers.
cfgte	slave	Signal	This signal provides default exception handling state.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.

Port	Direction	Protocol	Description
clrexmonack	master	Signal	Acknowledge handshake signal for the clrexmonreq signal
clrexmonreq	slave	Signal	Signals the clearing of an external global exclusive monitor
clusterid	slave	Value	The port sets the value of the affinity levels 1 and 2; bits [23:16] and [15:8] of the MPIDR.
CNTHPIRQ	master	Signal	Timer signals to SOC.
CNTPNSIRQ	master	Signal	Timer signals to SOC.
CNTPSIRQ	master	Signal	Timer signals to SOC.
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.
CNTVIRQ	master	Signal	Timer signals to SOC.
commirq	master	Signal	Interrupt signal from debug communications channel.
commr	master	Signal	Receive portion of Data Transfer Register full.
commt	master	Signal	Transmit portion of Data Transfer Register empty.
cp15sdisable2	slave	Signal	This signal disables write access to some system control processor registers.
cp15sdisable	slave	Signal	This signal disables write access to some system control processor registers.
cpuporeset	slave	Signal	CPU power on reset.
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
ctidbgirq	master	Signal	Cross trigger matrix port.
dbgack	master	Signal	External debug interface.
dbgen	slave	Signal	External debug interface.
dbgnopwrdown	master	Signal	No power-down request.
dbgpwrupreq	master	Signal	Processor powerup request.
dev_debug_s	slave	PVBus	External debug interface.
edbgrq	slave	Signal	External debug interface.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
l2flushdone	master	Signal	Flush of L2 memory system complete.
l2flushreq	slave	Signal	Request flush of L2 memory system.
l2reset	slave	Signal	Level2 reset.
memorymapped_debug_s	slave	PVBus	External debug interface.
niden	slave	Signal	External debug interface.
periphbase	slave	Value_64	This port sets the base address of private peripheral region.
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
presetdbg	slave	Signal	Debug reset.

Port	Direction	Protocol	Description
pvbus_m0	master	PVBus	The core will generate bus requests on this port.
rei	slave	Signal	Per core RAM Error Interrupt
reset	slave	Signal	Reset.
romaddr	slave	Value_64	Debug ROM base address.
romaddrv	slave	Signal	Debug ROM base address valid.
rvbaraddr	slave	Value_64	Reset vector base address.
sei	slave	Signal	Per core System Error physical pins.
smpen	master	Signal	This signals AMP or SMP mode for each core.
spiden	slave	Signal	External debug interface.
spniden	slave	Signal	External debug interface.
standbywfe	master	Signal	This signal indicates if a core is in WFE state.
standbywfi	master	Signal	This signal indicates if a core is in WFI state
standbywfil2	master	Signal	This signal indicated all cores and L2 are idles and in low power state.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfiq	slave	Signal	Virtualised FIQ.
vinithi	slave	Signal	This signal controls of the location of the exception vectors at reset.
virq	slave	Signal	Virtualised IRQ.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Per core virtual System Error physical pins.

## Parameters for ARMCortexA35CT

### **cpuX.AA64nAA32**

Type: `bool`

Default value: `true`

### **cpuX.CFGEND**

Type: `bool`

Default value: `false`

### **cpuX.CFGTE**

Type: `bool`

Default value: `false`

**cpuX.CP15SDISABLE**

Type: bool

Default value: false

**cpuX.CP15SDISABLE2**

Type: bool

Default value: false

**cpuX.CRYPTODISABLE**

Type: bool

Default value: false

**cpuX.RVBARADDR**

Type: uint64\_t

Default value: 0

**cpuX.VINITI**

Type: bool

Default value: false

**cpuX.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter trace\_special\_hlt\_imm16.

Type: bool

Default value: false

**cpuX.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint64\_t

Default value: 256

**cpuX.min\_sync\_level**

force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint32\_t

Default value: 0

**cpuX.semihosting-A32\_HLT**

Type: uint32\_t

Default value: 0xF000

**cpuX.semihosting-A64\_HLT**

Type: uint32\_t

Default value: 0xF000

**cpuX.semihosting-ARM\_SVC**

Type: uint32\_t

Default value: 0x123456

**cpuX.semihosting-T32\_HLT**

Type: uint32\_t

Default value: 0x3c

**cpuX.semihosting-Thumb\_SVC**

Type: uint32\_t

Default value: 0xAB

**cpuX.semihosting-cmd\_line**

Type: string

Default value: ""

**cpuX.semihosting-cwd**

Type: string

Default value: ""

**cpuX.semihosting-enable**

Type: bool

Default value: true

**cpuX.semihosting-heap\_base**

Type: uint32\_t

Default value: 0x0

**cpuX.semihosting-heap\_limit**

Type: uint32\_t

Default value: 0x0F000000

**cpuX.semihosting-stack\_base**

Type: uint32\_t

Default value: 0x10000000

**cpuX.semihosting-stack\_limit**

Type: uint32\_t

Default value: 0x0F000000

**cpuX.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**cpuX.vfp-enable\_at\_reset**

Type: bool

Default value: false

**cpuX.vfp-present**

Set whether the model has VFP support.

Type: bool

Default value: true

**BROADCASTCACHEMAINT**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

Type: bool

Default value: true

**BROADCASTINNER**

Enable broadcasting of Inner Shareable transactions. The broadcastinner signal will override this value if used.

Type: `bool`

Default value: `true`

### **BROADCASTOUTER**

Enable broadcasting of Outer Shareable transactions. The `broadcastouter` signal will override this value if used.

Type: `bool`

Default value: `true`

### **CLUSTER\_ID**

Type: `uint32_t`

Default value: 0

### **DBGROMADDR**

Type: `uint64_t`

Default value: `0x0`

### **DBGROMADDRV**

Type: `bool`

Default value: `true`

### **GICDISABLE**

Type: `bool`

Default value: `true`

### **NUM\_CORES**

Number of cores in cluster.

Type: `uint8_t`

Default value: 1

### **PERIPHBASE**

Type: `uint64_t`

Default value: `0x13080000`

### **bus\_type**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.



Type: `uint32_t`

Default value: 0

### **`cpi_div`**

divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: 1

### **`cpi_mul`**

multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: 1

### **`dcache-hit_latency`**

L1 D-Cache timing annotation latency for hit.

Type: `uint64_t`

Default value: 0

### **`dcache-maintenance_latency`**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

### **`dcache-miss_latency`**

L1 D-Cache timing annotation latency for miss.

Type: `uint64_t`

Default value: 0

### **`dcache-read_access_latency`**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (based on the size of `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

### **dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

### **dcache-size**

L1 D-Cache size in bytes.

Type: `uint32_t`

Default value: 0x8000

### **dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

### **dcache-state\_modelled**

Type: `bool`

Default value: `false`

### **dcache-write\_access\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (based on the size of `dcache-write_bus_width_in_bytes`. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

### **dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

**has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

**icache-hit\_latency**

L1 I-Cache timing annotation latency for hit.

Type: `uint64_t`

Default value: 0

**icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

**icache-miss\_latency**

L1 I-Cache timing annotation latency for miss.

Type: `uint64_t`

Default value: 0

**icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

**icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (based on the size of `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will

be used instead of per-byte even if `icache-read_latency` is set. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

### **`icache-read_latency`**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

### **`icache-size`**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: `0x8000`

### **`icache-state_modelled`**

Type: `bool`

Default value: `false`

### **`l2cache-hit_latency`**

L2 Cache timing annotation latency for hit.

Type: `uint64_t`

Default value: 0

### **`l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

### **`l2cache-miss_latency`**

L2 Cache timing annotation latency for miss.

Type: `uint64_t`

Default value: 0

**l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access (based on the size of l2cache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0

**l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0

**l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0

**l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0

**l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access (based on the size of l2cache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0

### **l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

### **ptw\_latency**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint32_t`

Default value: 0

### **tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint32_t`

Default value: 0

## 3.44 ARMCortexA510CT

Defined in `LISA/ARMCortexA510CT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support
r1p3	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following parameters were added:

- `has_delayed_dbgreg`
- `has_delayed_sysreg`
- `num_acp`

### About ARMCortexA510CT

The model supports the following features:

- DynamIQ Shared Unit-110 (DSU-110) system registers.
- L2 cache is supported at the per-core level only and there is no implementation of Core-Complex with shared L2 cache yet.
- PChannel for the cluster and for each core.
- `BROADCASTPERSIST` pin.
- Optional peripheral port.
- L3Cache partition.
- Per-core clock.
- Utility bus.
- Revision R1 is the default configuration, with 32-bit support at EL0. R1 supports both configurations of EL0, with or without A32 support. For 64-bit only mode, set parameter `max_32bit_el=-1`.
- To configure revision R0, set parameter `revision_number=0`.

Support for the following features is planned for a future release:

- `BROADCASTCACHEMAINTPOU` pin.
- `COREINSTRET`, `COREINSTRUN`, and `nPMBIRQ` signals.

The following features will not be implemented:

- DynamIQ features that are irrelevant to the programmers' view simulation, for example:
  - Automatic CPU retention mode.
  - Level-3 Cache RAM retention.
- 256-bit wide output transactions.
- Error correction or detection features.
- Self-test features (MBIST).
- Snoop filtering.
- Latency configuration.
- Cache stashing capability.

This model supports the Arm(R)v8-A Cryptographic Extensions, which requires the crypto plug-in (`crypto.dll` or `crypto.so`) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

## Iris and MTI instances for ARM Cortex A510CT

This model has the following Iris instances:

Name	Instance type
ARM Cortex A510CT	Cluster_ARM_Cortex-A510
ARM Cortex A510CT.AMU	PVBusLogger

Name	Instance type
ARMCortexA510CT.AMU.mapper	PVBusMapper
ARMCortexA510CT.DAP	PVBusLogger
ARMCortexA510CT.DAP.mapper	PVBusMapper
ARMCortexA510CT.DSU	DSU-110
ARMCortexA510CT.DSU.PPU_cluster	PPUv1
ARMCortexA510CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA510CT.DSU.PPU_core0	PPUv1
ARMCortexA510CT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexA510CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA510CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA510CT.DSU.shared_cache	PVCache
ARMCortexA510CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA510CT.DSU.shared_cache.upstream[Y] (where Y = 0-3)	PVBusSlave
ARMCortexA510CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA510CT.MMAP	PVBusLogger
ARMCortexA510CT.MMAP.mapper	PVBusMapper
ARMCortexA510CT.RAS	PVBusLogger
ARMCortexA510CT.RAS.mapper	PVBusMapper
ARMCortexA510CT.cpu0	ARM_Cortex-A510
ARMCortexA510CT.cpu0.UTLB	TLB
ARMCortexA510CT.cpu0.debug_rom	debug_rom
ARMCortexA510CT.cpu0.dtlb	TLB
ARMCortexA510CT.cpu0.l1dcache	PVCache
ARMCortexA510CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA510CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA510CT.cpu0.l1icache	PVCache
ARMCortexA510CT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA510CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA510CT.cpu0.l2cache	PVCache
ARMCortexA510CT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA510CT.cpu0.l2cache.upstream[U] (where U = 0-1)	PVBusSlave
ARMCortexA510CT.ext_bus	PVBusLogger
ARMCortexA510CT.ext_bus.mapper	PVBusMapper
ARMCortexA510CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA510CT.global_debug_rom	debug_rom
ARMCortexA510CT.secondary_debug_rom	debug_rom
ARMCortexA510CT.sve	ScalableVectorExtension

This model has the following MTI trace components:



Name	Component type
ARMCortexA510CT.AMU	PVBusLogger
ARMCortexA510CT.AMU.mapper	PVBusMapper
ARMCortexA510CT.DAP	PVBusLogger
ARMCortexA510CT.DAP.mapper	PVBusMapper
ARMCortexA510CT.DSU	DSU-110
ARMCortexA510CT.DSU.PPU_cluster	PPUv1
ARMCortexA510CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA510CT.DSU.PPU_core0	PPUv1
ARMCortexA510CT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexA510CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA510CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA510CT.DSU.shared_cache	PVCache
ARMCortexA510CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA510CT.DSU.shared_cache.upstream[Y] (where Y = 0-3)	PVBusSlave
ARMCortexA510CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA510CT.MMAP	PVBusLogger
ARMCortexA510CT.MMAP.mapper	PVBusMapper
ARMCortexA510CT.RAS	PVBusLogger
ARMCortexA510CT.RAS.mapper	PVBusMapper
ARMCortexA510CT.cpu0	ARM_Cortex-A510
ARMCortexA510CT.cpu0.UTLB	TLB
ARMCortexA510CT.cpu0.l1dcache	PVCache
ARMCortexA510CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA510CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA510CT.cpu0.l1icache	PVCache
ARMCortexA510CT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA510CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA510CT.cpu0.l2cache	PVCache
ARMCortexA510CT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA510CT.cpu0.l2cache.upstream[U] (where U = 0-1)	PVBusSlave
ARMCortexA510CT.ext_bus	PVBusLogger
ARMCortexA510CT.ext_bus.mapper	PVBusMapper
ARMCortexA510CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

## Ports for ARMCortexA510CT

Port	Direction	Protocol	Description
acp_s	slave	PVBus	AXI ACP slave port.
AENDOMP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).

Port	Direction	Protocol	Description
AEND2MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND3MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART2MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	slave	Signal	CHI defined pins.
broadcastcachemaint	slave	Signal	ACE defined pins.
broadcastouter	slave	Signal	ACE defined pins.
broadcastpersist	slave	Signal	CHI defined pins.
cfgend	slave	Signal	This signal is for EE bit initialisation.
cfgsdisable	slave	Signal	This signal disables write access to some secure Interrupt Controller registers.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcsmpchannel	master	PChannel	Cluster PCSM signal
cluster_powerdown_out	master	Signal	DEPRECATED - An output from PPU that informs thermal controller of the cluster power info
cluster_ppu_hw_stat	master	Value	Cluster PPU power state
clusterid	slave	Value	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set ManagerID64 of the core.
clusterpmuirq	master	Signal	DynamiQ pmu irq
CNTHPIRQ	master	Signal	Timer signals to SOC.
CNTHPSIRQ	master	Signal	Timer signals to SOC
CNTHVIRQ	master	Signal	Timer signals to SOC.
CNTHVSIRQ	master	Signal	Timer signals to SOC
CNTPNSIRQ	master	Signal	Timer signals to SOC.
CNTPSIRQ	master	Signal	Timer signals to SOC.
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.

Port	Direction	Protocol	Description
CNTVIRQ	master	Signal	Timer signals to SOC.
commirq	master	Signal	Interrupt signal from debug communications channel.
core_clk_in	slave	ClockSignal	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
core_pcsn_pchannel	master	PChannel	Core PCSM signals
core_powerdown_out	master	Signal	DEPRECATED - An output from PPUs that informs thermal controller of the core power info
core_ppu_hw_stat	master	Value	Notify the power state change inside the PPU.
coreerrirq	master	Signal	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq	master	Signal	Fault indicator for a detected 1 or 2 bit ECC error
coreinstrrun	master	Signal	Core Running State
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
ctidbgirq	master	Signal	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	slave	Signal	External debug interface.
dbgnopwrdown	master	Signal	No power-down request.
dbgprupreq	master	Signal	Debug power up request.
defaultMP	slave	Signal	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main master interface ports.
dev_debug_s	slave	PVBus	External debug interface.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.
gicreset	master	Signal	An output from PPUs that can be used to reset GICCLK domain components
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	slave	PVBus	External debug interface.
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	master	Signal	PPU cluster interrupt.
ppu_cluster_isolate	master	Signal	PPU Cluster Domain isolation control.
ppu_cluster_wakerequest	slave	Signal	PPU cluster wake request signal.
ppu_core_irq	master	Signal	PPU core interrupt.
ppu_core_isolate	master	Signal	PPU Core Domain isolation control.
ppu_core_wakerequest	slave	Signal	PPU Core wake request signals.
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	master	PVBus	The core will generate bus requests on this port.

Port	Direction	Protocol	Description
pvbus_periph_m	master	PVBus	The core can generate peripheral bus request on this port.
rei	slave	Signal	Per core RAM Error Interrupt.
reset	slave	Signal	Reset signal to cluster.
rvbaraddr	slave	Value_64	Reset vector base address.
sci_m	master	SystemCoherencyInterface	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei	slave	Signal	Per core System Error physical pins.
spiden	slave	Signal	External debug interface.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq	master	Signal	Interrupt signal from the trace buffer unit.
utility_bus_s	slave	PVBus	Utility bus slave
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfiq	slave	Signal	Virtualised FIQ.
virq	slave	Signal	Virtualised IRQ.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Per core virtual System Error physical pins.

## Parameters for ARM Cortex A510CT

### cpuX.CFGEND

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### cpuX.CRYPTODISABLE

Disable cryptographic features.

Type: `bool`

Default value: `false`

### cpuX.RVBARADDR

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**cpuX.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter trace\_special\_hlt\_imm16.

Type: `bool`

Default value: `false`

**cpuX.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**cpuX.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**cpuX.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**cpuX.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**cpuX.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: `uint32_t`

Default value: `0x10`

**cpuX.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**cpuX.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**cpuX.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**cpuX.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**cpuX.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**cpuX.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**cpuX.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**cpuX.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

### **`cpuX.semihosting-T32_HLT`**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

### **`cpuX.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

### **`cpuX.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

### **`cpuX.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

### **`cpuX.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`cpuX.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.semihosting-heap_limit`**

Virtual address of top of heap.



Type: `uint64_t`

Default value: `0xf0000000`

### **`cpuX.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`cpuX.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf0000000`

### **`cpuX.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`cpuX.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`cpuX.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`AEND0_DEFAULT`**

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMPO input signal).

Type: `uint64_t`

Default value: 0x0

#### **AEND1\_DEFAULT**

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

Type: uint64\_t

Default value: 0x0

#### **AEND2\_DEFAULT**

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

Type: uint64\_t

Default value: 0x0

#### **AEND3\_DEFAULT**

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

Type: uint64\_t

Default value: 0x0

#### **ASTART0\_DEFAULT**

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

Type: uint64\_t

Default value: 0x0

#### **ASTART1\_DEFAULT**

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

Type: uint64\_t

Default value: 0x0

#### **ASTART2\_DEFAULT**

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

Type: uint64\_t

Default value: `0x0`

### **ASTART3\_DEFAULT**

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

Type: `uint64_t`

Default value: `0x0`

### **BROADCASTATOMIC**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`

Default value: `true`

### **BROADCASTCACHEMAINT**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

Type: `bool`

Default value: `false`

### **BROADCASTOUTER**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

Type: `bool`

Default value: `false`

### **BROADCASTPERSIST**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: `bool`

Default value: `true`

### **CLUSTER\_ID**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1.

This configuration also updates all relevant component DEVAFF registers and is used to set the ManagerID64 of the core.

Type: `uint32_t`

Default value: `0x0`

### **CMO\_broadcast\_when\_cache\_state\_modelling\_disabled**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. This value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required.

Type: `uint8_t`

Default value: `1`

### **CPUCFR**

Value of CPU Configuration Register.

Type: `uint64_t`

Default value: `0x0`

### **GICDISABLE**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`

Default value: `true`

### **NUM\_CORES**

Number of cores per cluster.

Type: `uint8_t`

Default value: `1`

### **bus\_type**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

Type: `uint8_t`

Default value: `0`

**core\_power\_on\_by\_default**

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPUs which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

Type: `bool`

Default value: `false`

**cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-maintenance\_latency**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-miss\_latency**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-prefetch\_enabled**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

Type: `bool`

Default value: `false`

**dcache-read\_access\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**dcache-size**

L1 D-Cache size in bytes.

Type: `uint32_t`

Default value: `0x10000`

**dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**dcache-state\_modelled**

Set whether D-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **`dcache-write_access_latency`**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`dcache-write_latency`**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`enable_simulation_performance_optimizations`**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of `stage12_tlb_size` parameter to 1024).

Type: `bool`

Default value: `true`

### **`ete.CLAIMTAGS`**

Number of claim tags.

Type: `uint8_t`

Default value: 4

### **`ete.MAX_INST_PER_Q`**

Maximum limit for the number of instructions implied by a Q element.

Type: `uint16_t`

Default value: `0x1`

### **`ete.NumberOfRSPairs`**

Number of resource selector pairs.

Type: `uint8_t`

Default value: 8

#### **ete.PIDR\_CM0D**

TRCPIDR CM0D value.

Type: `uint8_t`

Default value: 0

#### **ete.PIDR\_REVAND**

TRCPIDR REVAND value.

Type: `uint8_t`

Default value: 0

#### **ete.PIDR\_REVISION**

TRCPIDR REVISION value.

Type: `uint8_t`

Default value: 0

#### **ete.Q\_CADENCE**

Number of instruction blocks traced between two Q elements.

Type: `uint16_t`

Default value: 0x1

#### **ete.RES0\_STATEFUL**

Whether **RES0** bits are stateful or **RAZ/WI**.

Type: `bool`

Default value: `false`

#### **ete.RETSTACK**

Return stack depth.

Type: `uint8_t`

Default value: 3

#### **ete.REVISION**

TRCIDR1 revision value.



Type: `uint8_t`

Default value: 0

### **`ete.SIM_OVERFLOW_GRANULARITY`**

Number of instruction blocks in each granule, for simulated overflow.

Type: `uint32_t`

Default value: 0x64

### **`ete.SIM_OVERFLOW_PERCENTAGE`**

Percentage of instruction blocks lost in each granule, for simulated overflow.

Type: `uint8_t`

Default value: 0

### **`ete.SOURCE_ADDRESS`**

Allow generation of source address elements.

Type: `bool`

Default value: `false`

### **`ete.TRACE_OUTPUT`**

File to which to write trace byte stream.

Type: `string`

Default value: N/A

### **`ete.TRCSRSTA_FORCED_EXCEP`**

TRCSRSTA value for a forcibly traced exception.

Type: `bool`

Default value: `false`

### **`force_mte_tag_access_razwi_and_ignore_tag_checks`**

Force MTE tag accesses to **RAZ/WI** and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

Type: `bool`

Default value: `false`

**force\_zero\_PSTATE\_PAN**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

**force\_zero\_mpam\_partid\_and\_pmg**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

Type: `uint8_t`

Default value: `0`

**has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

**has\_dot\_product**

Possible values of this parameter are:- 1, feature not implemented.- 2, feature is implemented.

Type: `uint8_t`

Default value: `2`

**has\_ete**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (`-plugin` or `-P`).

Type: `bool`

Default value: `false`

**has\_peripheral\_port**

If true, an additional AXI peripheral port is configured.

Type: `bool`

Default value: `false`

**icache-hit\_latency**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

**icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **icache-size**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: 0x10000

### **icache-state\_modelled**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **invalidate\_code\_cache\_on\_icache\_cmo**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: `uint8_t`

Default value: 0

### **l3cache-has\_mpam**

L3 Cache has MPAM support.

Type: `bool`

Default value: `true`

### **l3cache-hit\_latency**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**l3cache-maintenance\_latency**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-miss\_latency**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-read\_access\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size `l3cache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-read\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l3cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-size**

L3 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

**l3cache-snoop\_data\_transfer\_latency**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-snoop_issue_latency`**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-ways`**

L3 Cache number of ways (sets are implicit from size).

Type: `uint16_t`

Default value: `16`

### **`l3cache-write_access_latency`**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size `l3cache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-write_latency`**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l3cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`max_32bit_el`**

Maximum exception level supporting AArch32 modes. -1: No Support for A32 at any EL, 0: EL0 supports A32. This parameter is ignored in Rev0 i.e. when `revision_number = 0`.

Type: `int8_t`

Default value: `0`

**memory\_tagging\_support\_level**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

Type: `uint8_t`

Default value: 3

**mpmm\_accumulator\_multiplier**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator.value of n means the accumulator will use (n \* accumulator value) to calculate the mpmm threshold (MPMM).is provided as a fast model workaround to handle cases where execution of quantum in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

Type: `uint8_t`

Default value: 1

**num\_acp**

Number of ACP ports.

Type: `uint8_t`

Default value: 0

**num\_nodes**

Number of transport nodes. Zero implies direct-connect configuration.

Type: `uint8_t`

Default value: 1

**patch\_level**

Patch level of TRM implemented by the model. Changing the value can change the model behaviour. Visible as the Revision field in MIDR/MIDR\_EL1. Corresponds to the patch number Y in rXpY.

Type: `uint8_t`

Default value: 3

**ptw\_latency**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

### **ras\_pfg\_clock\_mhz**

RAS Pseudo-Fault generation clock rate in MHz.

Type: `uint8_t`

Default value: 12

### **revision\_number**

Revision number of TRM implemented by the model. Changing the value can change the model behaviour. Visible as the Variant field in MIDR/MIDR\_EL1. Corresponds to the revision number X in rXpY.

Type: `uint8_t`

Default value: 1

### **tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

### **tlbi\_stall\_enabled**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: `bool`

Default value: `false`

### **treat\_PAC\_as\_NOP**

Non-architecture parameter. Treat Pointer Authentication as **NOP**. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are **NOP**. Branch, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

Type: `bool`

Default value: `false`

### **walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`



Default value: 0x0

## 3.45 ARMCortexA510CT\_CortexA710CT

Defined in LISA/ARMCortexA510CT\_CortexA710CT.lisa.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
CortexA510 r0p0	Full support
CortexA510 r1p3	Full support
CortexA710 r2p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following parameters were added:

- num\_acp
- subcluster0.has\_delayed\_dbgreg
- subcluster0.has\_delayed\_sysreg
- subcluster1.has\_delayed\_dbgreg
- subcluster1.has\_delayed\_sysreg

### About ARMCortexA510CT\_CortexA710CT

The number of cores in each subcluster is configurable using the following parameters:

#### **subcluster0.NUM\_CORES**

Possible values are 1-11 (ARMCortexA510CT).

#### **subcluster1.NUM\_CORES**

Possible values are 1-11 (ARMCortexA710CT).

The total number of cores in the cluster cannot exceed 12.

Port arrays are expanded to the width needed for the maximum number of cores in each subcluster. Use the following port array indexes:

- <port\_name>[0-10] for cores in subcluster0.
- <port\_name>[11-21] for cores in subcluster1.



All instances in the Master cross trigger matrix port array `cti[22]` must be connected, regardless of the `NUM_CORES` value used.

Core-specific parameters have the following prefixes:

- `subcluster0.cpu0` to `subcluster0.cpu10` identify cores in `subcluster0`.
- `subcluster1.cpu0` to `subcluster1.cpu10` identify cores in `subcluster1`.

For information about the cores in this model, see:

- [ARMCortexA510CT](#).
- [ARMCortexA710CT](#).

## Iris and MTI instances for ARMCortexA510CT\_CortexA710CT

This model has the following Iris instances:

Name	Instance type
ARMCortexA510CT_CortexA710CT	Cluster_ARM_Cortex510_CortexA710_Heterogeneous
ARMCortexA510CT_CortexA710CT.AMU	PVBusLogger
ARMCortexA510CT_CortexA710CT.AMU.mapper	PVBusMapper
ARMCortexA510CT_CortexA710CT.DAP	PVBusLogger
ARMCortexA510CT_CortexA710CT.DAP.mapper	PVBusMapper
ARMCortexA510CT_CortexA710CT.DSU	DSU-110
ARMCortexA510CT_CortexA710CT.DSU.PPU_cluster	PPUv1
ARMCortexA510CT_CortexA710CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA510CT_CortexA710CT.DSU.PPU_coreZ (where Z = 0-1)	PPUv1
ARMCortexA510CT_CortexA710CT.DSU.PPU_coreZ.busslave (where Z = 0-1)	PVBusSlave
ARMCortexA510CT_CortexA710CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA510CT_CortexA710CT.DSU.shared_cache	PVCache
ARMCortexA510CT_CortexA710CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA510CT_CortexA710CT.DSU.shared_cache.upstream[Z] (where Z = 0-5)	PVBusSlave
ARMCortexA510CT_CortexA710CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT.MMAP	PVBusLogger
ARMCortexA510CT_CortexA710CT.MMAP.mapper	PVBusMapper
ARMCortexA510CT_CortexA710CT.RAS	PVBusLogger
ARMCortexA510CT_CortexA710CT.RAS.mapper	PVBusMapper
ARMCortexA510CT_CortexA710CT.cpuZ.debug_rom (where Z = 0-1)	debug_rom
ARMCortexA510CT_CortexA710CT.ext_bus	PVBusLogger
ARMCortexA510CT_CortexA710CT.ext_bus.mapper	PVBusMapper
ARMCortexA510CT_CortexA710CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA510CT_CortexA710CT.global_debug_rom	debug_rom
ARMCortexA510CT_CortexA710CT.secondary_debug_rom	debug_rom
ARMCortexA510CT_CortexA710CT.subcluster0	Subcluster_ARM_Cortex-A510
ARMCortexA510CT_CortexA710CT.subcluster0.cpu0	ARM_Cortex-A510

Name	Instance type
ARMCortexA510CT_CortexA710CT.subclusterZ.cpu0.UTLB (where Z = 0-1)	TLB
ARMCortexA510CT_CortexA710CT.subclusterZ.cpuU.dtlb (where Z = 0-1; U = 0-1)	TLB
ARMCortexA510CT_CortexA710CT.subclusterZ.cpu0.l1dcache (where Z = 0-1)	PVCache
ARMCortexA510CT_CortexA710CT.subclusterZ.cpu0.l1dcache.downstream[0].pvbusmaster (where Z = 0-1)	PVBusMaster
ARMCortexA510CT_CortexA710CT.subclusterZ.cpu0.l1dcache.upstream[0] (where Z = 0-1)	PVBusSlave
ARMCortexA510CT_CortexA710CT.subclusterZ.cpu0.l1icache (where Z = 0-1)	PVCache
ARMCortexA510CT_CortexA710CT.subclusterZ.cpu0.l1icache.downstream[0].pvbusmaster (where Z = 0-1)	PVBusMaster
ARMCortexA510CT_CortexA710CT.subclusterZ.cpu0.l1icache.upstream[0] (where Z = 0-1)	PVBusSlave
ARMCortexA510CT_CortexA710CT.subclusterZ.cpu0.l2cache (where Z = 0-1)	PVCache
ARMCortexA510CT_CortexA710CT.subclusterZ.cpu0.l2cache.downstream[0].pvbusmaster (where Z = 0-1)	PVBusMaster
ARMCortexA510CT_CortexA710CT.subclusterZ.cpu0.l2cache.upstream[W] (where Z = 0-1; W = 0-1)	PVBusSlave
ARMCortexA510CT_CortexA710CT.subclusterZ.sve (where Z = 0-1)	ScalableVectorExtension
ARMCortexA510CT_CortexA710CT.subcluster1	Subcluster_ARM_Cortex-A710
ARMCortexA510CT_CortexA710CT.subcluster1.cpu0	ARM_Cortex-A710

This model has the following MTI trace components:

Name	Component type
ARMCortexA510CT_CortexA710CT.AMU	PVBusLogger
ARMCortexA510CT_CortexA710CT.AMU.mapper	PVBusMapper
ARMCortexA510CT_CortexA710CT.DAP	PVBusLogger
ARMCortexA510CT_CortexA710CT.DAP.mapper	PVBusMapper
ARMCortexA510CT_CortexA710CT.DSU	DSU-110
ARMCortexA510CT_CortexA710CT.DSU.PPU_cluster	PPUv1
ARMCortexA510CT_CortexA710CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA510CT_CortexA710CT.DSU.PPU_coreZ (where Z = 0-1)	PPUv1
ARMCortexA510CT_CortexA710CT.DSU.PPU_coreZ.busslave (where Z = 0-1)	PVBusSlave
ARMCortexA510CT_CortexA710CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA510CT_CortexA710CT.DSU.shared_cache	PVCache
ARMCortexA510CT_CortexA710CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA510CT_CortexA710CT.DSU.shared_cache.upstream[Z] (where Z = 0-5)	PVBusSlave
ARMCortexA510CT_CortexA710CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT.MMAP	PVBusLogger

Name	Component type
ARMCortexA510CT_CortexA710CT.MMAP.mapper	PVBusMapper
ARMCortexA510CT_CortexA710CT.RAS	PVBusLogger
ARMCortexA510CT_CortexA710CT.RAS.mapper	PVBusMapper
ARMCortexA510CT_CortexA710CT.ext_bus	PVBusLogger
ARMCortexA510CT_CortexA710CT.ext_bus.mapper	PVBusMapper
ARMCortexA510CT_CortexA710CT.gic_cpuif_decoder_cluster	GlCv3CPUInterfaceDecoder
ARMCortexA510CT_CortexA710CT.subcluster0.cpu0	ARM_Cortex-A510
ARMCortexA510CT_CortexA710CT.subclusterZ.cpu0.UTLB (where Z = 0-1)	TLB
ARMCortexA510CT_CortexA710CT.subclusterZ.cpu0.l1dcache (where Z = 0-1)	PVCache
ARMCortexA510CT_CortexA710CT.subclusterZ.cpu0.l1dcache.downstream[0].pvbusmaster (where Z = 0-1)	PVBusMaster
ARMCortexA510CT_CortexA710CT.subclusterZ.cpu0.l1dcache.upstream[0] (where Z = 0-1)	PVBusSlave
ARMCortexA510CT_CortexA710CT.subclusterZ.cpu0.l1icache (where Z = 0-1)	PVCache
ARMCortexA510CT_CortexA710CT.subclusterZ.cpu0.l1icache.downstream[0].pvbusmaster (where Z = 0-1)	PVBusMaster
ARMCortexA510CT_CortexA710CT.subclusterZ.cpu0.l1icache.upstream[0] (where Z = 0-1)	PVBusSlave
ARMCortexA510CT_CortexA710CT.subclusterZ.cpu0.l2cache (where Z = 0-1)	PVCache
ARMCortexA510CT_CortexA710CT.subclusterZ.cpu0.l2cache.downstream[0].pvbusmaster (where Z = 0-1)	PVBusMaster
ARMCortexA510CT_CortexA710CT.subclusterZ.cpu0.l2cache.upstream[W] (where Z = 0-1; W = 0-1)	PVBusSlave
ARMCortexA510CT_CortexA710CT.subcluster1.cpu0	ARM_Cortex-A710

### Ports for ARMCortexA510CT\_CortexA710CT

Port	Direction	Protocol	Description
acp_s	slave	PVBus	AXI ACP slave port.
AEND0MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND2MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND3MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART2MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).

Port	Direction	Protocol	Description
broadcastatomic	slave	Signal	CHI defined pins.
broadcastcachemaint	slave	Signal	ACE defined pins.
broadcastouter	slave	Signal	ACE defined pins.
broadcastpersist	slave	Signal	CHI defined pins.
cfgend	slave	Signal	This signal is for EE bit initialisation.
cfgsdisable	slave	Signal	This signal disables write access to some secure Interrupt Controller registers.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcs_m_pchannel	master	PChannel	Cluster PCSM signal
cluster_powerdown_out	master	Signal	DEPRECATED - An output from PPU that informs thermal controller of the cluster power info
cluster_ppu_hw_stat	master	Value	Cluster PPU power state
clusterid	slave	Value	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set ManagerID64 of the core.
clusterpmuirq	master	Signal	DynamiQ pmu irq
CNTHPIRQ	master	Signal	Timer signals to SOC.
CNTHPSIRQ	master	Signal	Timer signals to SOC
CNTHVIRQ	master	Signal	Timer signals to SOC.
CNTHVSIQ	master	Signal	Timer signals to SOC
CNTPNSIRQ	master	Signal	Timer signals to SOC.
CNTPSIQ	master	Signal	Timer signals to SOC.
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.
CNTVIRQ	master	Signal	Timer signals to SOC.
commirq	master	Signal	Interrupt signal from debug communications channel.
core_clk_in	slave	ClockSignal	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
core_pcs_m_pchannel	master	PChannel	Core PCSM signals
core_powerdown_out	master	Signal	DEPRECATED - An output from PPU that informs thermal controller of the core power info
core_ppu_hw_stat	master	Value	Notify the power state change inside the PPU.
coreerrirq	master	Signal	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq	master	Signal	Fault indicator for a detected 1 or 2 bit ECC error

Port	Direction	Protocol	Description
coreinstrrun	master	Signal	Core Running State
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
ctidbgirq	master	Signal	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	slave	Signal	External debug interface.
dbgnopwrdown	master	Signal	No power-down request.
dbgpwrupreq	master	Signal	Debug power up request.
defaultMP	slave	Signal	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main master interface ports.
dev_debug_s	slave	PVBus	External debug interface.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.
gicreset	master	Signal	An output from PPU that can be used to reset GICCLK domain components
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	slave	PVBus	External debug interface.
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	master	Signal	PPU cluster interrupt.
ppu_cluster_isolate	master	Signal	PPU Cluster Domain isolation control.
ppu_cluster_wakerequest	slave	Signal	PPU cluster wake request signal.
ppu_core_irq	master	Signal	PPU core interrupt.
ppu_core_isolate	master	Signal	PPU Core Domain isolation control.
ppu_core_wakerequest	slave	Signal	PPU Core wake request signals.
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	master	PVBus	The core will generate bus requests on this port.
pvbus_periph_m	master	PVBus	The core can generate peripheral bus request on this port.
rei	slave	Signal	Per core RAM Error Interrupt.
reset	slave	Signal	Reset signal to cluster.
rvbaraddr	slave	Value_64	Reset vector base address.
sci_m	master	SystemCoherencyInterface	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei	slave	Signal	Per core System Error physical pins.
spiden	slave	Signal	External debug interface.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.

Port	Direction	Protocol	Description
trbirq	master	Signal	Interrupt signal from the trace buffer unit.
utility_bus_s	slave	PVBus	Utility bus slave
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfiq	slave	Signal	Virtualised FIQ.
virq	slave	Signal	Virtualised IRQ.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Per core virtual System Error physical pins.

## Parameters for ARMCortexA510CT\_CortexA710CT

### **AEND0\_DEFAULT**

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

Type: uint64\_t

Default value: 0x0

### **AEND1\_DEFAULT**

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

Type: uint64\_t

Default value: 0x0

### **AEND2\_DEFAULT**

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

Type: uint64\_t

Default value: 0x0

### **AEND3\_DEFAULT**

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

Type: uint64\_t

Default value: 0x0

**ASTART0\_DEFAULT**

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART1\_DEFAULT**

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART2\_DEFAULT**

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART3\_DEFAULT**

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

Type: `uint64_t`

Default value: `0x0`

**BROADCASTATOMIC**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`

Default value: `true`

**BROADCASTCACHEMAINT**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

Type: `bool`

Default value: `true`



**BROADCASTOUTER**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

Type: `bool`

Default value: `true`

**BROADCASTPERSIST**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: `bool`

Default value: `true`

**CLUSTER\_ID**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the ManagerID64 of the core.

Type: `uint32_t`

Default value: `0x0`

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. This value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required.

Type: `uint8_t`

Default value: `1`

**GICDISABLE**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`

Default value: `true`

**core\_power\_on\_by\_default**

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPUs which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

Type: `bool`

Default value: `false`

**dcache-state\_modelled**

Set whether D-cache has stateful implementation.

Type: `bool`

Default value: `false`

**default\_opmode**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

Type: `uint8_t`

Default value: 4

**diagnostics**

Enable DynamIQ diagnostic messages.

Type: `bool`

Default value: `false`

**enable\_simulation\_performance\_optimizations**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of `stage12_tlb_size` parameter to 1024).

Type: `bool`

Default value: `true`

**has\_acp**

If true, Accelerator Coherency Port is configured.

Type: `bool`

Default value: `false`

**has\_peripheral\_port**

If true, an additional AXI peripheral port is configured.

Type: `bool`

Default value: `false`

**icache-state\_modelled**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

**l3cache-has\_mpam**

L3 Cache has MPAM support.

Type: `bool`

Default value: `false`

**l3cache-hit\_latency**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-maintenance\_latency**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-miss\_latency**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-read\_access\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size l3cache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**l3cache-read\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. l3cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**l3cache-size**

L3 Cache size in bytes.

Type: uint32\_t

Default value: 0x0

**l3cache-snoop\_data\_transfer\_latency**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**l3cache-snoop\_issue\_latency**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**l3cache-write\_access\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size l3cache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used

instead of per-byte even if l3cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **l3cache-write\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. l3cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **mpmm\_accumulator\_multiplier**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator.value of n means the accumulator will use (n \* accumulator value) to calculate the mpmm threshold (MPMM).is provided as a fast model workaround to handle cases where execution of quantum in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

Type: uint8\_t

Default value: 1

### **num\_acp**

Number of ACP ports.

Type: uint8\_t

Default value: 0

### **num\_nodes**

Number of transport nodes. Zero implies direct-connect configuration.

Type: uint8\_t

Default value: 1

### **periph\_address\_end**

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

Type: uint64\_t

Default value: 0x0

**periph\_address\_start**

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

Type: `uint64_t`

Default value: `0x0`

**subcluster0.CPUCFR**

Value of CPU Configuration Register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.NUM\_CORES**

Number of cores in the subcluster. Total number of cores in cluster may not exceed 12.

Type: `uint8_t`

Default value: `1`

**subcluster0.cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**subcluster0.cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**subcluster0.cpu0.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster0.cpu0.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu0.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu0.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu0.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu0.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu0.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu0.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster0.cpu0.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu0.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu0.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8



**subcluster0.cpu0.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu0.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu0.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu0.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu0.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu0.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu0.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster0.cpu0.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster0.cpu0.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu0.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu0.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu0.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu0.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu0.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu0.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu0.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu0.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster0.cpu0.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu1.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu1.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu1.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu1.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu1.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu1.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu1.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu1.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu1.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu1.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu1.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu1.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu1.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu1.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu1.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu1.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster0.cpu1.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster0.cpu1.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster0.cpu1.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster0.cpu1.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu1.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu1.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu1.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu1.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu1.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.



Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster0.cpu1.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu1.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster0.cpu10.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu10.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu10.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu10.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu10.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu10.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu10.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu10.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu10.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu10.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu10.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu10.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu10.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu10.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu10.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu10.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu10.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu10.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu10.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu10.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu10.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu10.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu10.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

### **`subcluster0.cpu10.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

### **`subcluster0.cpu10.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **`subcluster0.cpu10.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **`subcluster0.cpu10.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster0.cpu10.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu10.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu10.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster0.cpu10.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu10.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu10.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu10.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu2.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu2.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu2.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu2.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu2.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu2.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu2.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu2.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster0.cpu2.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu2.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu2.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8



**subcluster0.cpu2.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu2.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu2.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu2.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu2.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu2.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu2.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster0.cpu2.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster0.cpu2.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu2.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu2.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu2.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu2.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu2.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu2.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu2.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu2.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster0.cpu2.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **subcluster0.cpu3.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **subcluster0.cpu3.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **subcluster0.cpu3.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **subcluster0.cpu3.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **subcluster0.cpu3.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **subcluster0.cpu3.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu3.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu3.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu3.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu3.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu3.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu3.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu3.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu3.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu3.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu3.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster0.cpu3.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster0.cpu3.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster0.cpu3.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster0.cpu3.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu3.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu3.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu3.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu3.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu3.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.



Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster0.cpu3.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu3.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster0.cpu4.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu4.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu4.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu4.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu4.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu4.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu4.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

#### **subcluster0.cpu4.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu4.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu4.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

#### **subcluster0.cpu4.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu4.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu4.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu4.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu4.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu4.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu4.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu4.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

### **`subcluster0.cpu4.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

### **`subcluster0.cpu4.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **`subcluster0.cpu4.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **`subcluster0.cpu4.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster0.cpu4.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu4.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu4.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

#### **`subcluster0.cpu4.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

#### **`subcluster0.cpu4.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster0.cpu4.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu4.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster0.cpu5.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu5.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu5.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu5.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu5.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu5.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu5.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu5.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster0.cpu5.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu5.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu5.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8



**subcluster0.cpu5.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu5.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu5.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu5.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu5.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu5.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu5.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster0.cpu5.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster0.cpu5.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu5.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu5.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu5.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu5.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu5.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu5.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu5.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu5.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster0.cpu5.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu6.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu6.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu6.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu6.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu6.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu6.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu6.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu6.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu6.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu6.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu6.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu6.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu6.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu6.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu6.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu6.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster0.cpu6.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster0.cpu6.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster0.cpu6.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster0.cpu6.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu6.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu6.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu6.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu6.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu6.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.



Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster0.cpu6.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu6.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster0.cpu7.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu7.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu7.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu7.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu7.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu7.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu7.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu7.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu7.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu7.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu7.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu7.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu7.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu7.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**`subcluster0.cpu7.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**`subcluster0.cpu7.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**`subcluster0.cpu7.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**`subcluster0.cpu7.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**`subcluster0.cpu7.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**`subcluster0.cpu7.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**`subcluster0.cpu7.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster0.cpu7.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu7.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu7.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu7.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu8.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu8.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu8.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu8.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu8.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu8.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu8.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu8.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster0.cpu8.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu8.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu8.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8



**subcluster0.cpu8.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu8.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu8.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu8.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu8.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu8.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu8.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster0.cpu8.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster0.cpu8.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu8.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu8.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu8.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu8.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu8.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu8.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu8.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu8.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster0.cpu8.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **subcluster0.cpu9.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **subcluster0.cpu9.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **subcluster0.cpu9.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **subcluster0.cpu9.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **subcluster0.cpu9.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **subcluster0.cpu9.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu9.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu9.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu9.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu9.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu9.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu9.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu9.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu9.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu9.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu9.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster0.cpu9.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster0.cpu9.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster0.cpu9.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster0.cpu9.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu9.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu9.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu9.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu9.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu9.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.



Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster0.cpu9.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu9.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster0.dcache-hit_latency`**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.dcache-maintenance_latency`**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.dcache-miss_latency`**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.dcache-prefetch_enabled`**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

Type: `bool`

Default value: `false`

### **`subcluster0.dcache-read_access_latency`**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.dcache-read_latency`**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.dcache-size`**

L1 D-Cache size in bytes.

Type: `uint32_t`

Default value: `0x10000`

### **`subcluster0.dcache-snoop_data_transfer_latency`**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.dcache-write_access_latency`**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.ete.CLAIMTAGS**

Number of claim tags.

Type: uint8\_t

Default value: 4

**subcluster0.ete.MAX\_INST\_PER\_Q**

Maximum limit for the number of instructions implied by a Q element.

Type: uint16\_t

Default value: 0x1

**subcluster0.ete.NumberOfRSPairs**

Number of resource selector pairs.

Type: uint8\_t

Default value: 8

**subcluster0.ete.PIDR\_CM0D**

TRCPIDR CM0D value.

Type: uint8\_t

Default value: 0

**subcluster0.ete.PIDR\_REVAND**

TRCPIDR REVAND value.

Type: uint8\_t

Default value: 0

**subcluster0.ete.PIDR\_REVISION**

TRCPIDR REVISION value.

Type: `uint8_t`

Default value: 0

#### **`subcluster0.ete.Q_CADENCE`**

Number of instruction blocks traced between two Q elements.

Type: `uint16_t`

Default value: 0x1

#### **`subcluster0.ete.RES0_STATEFUL`**

Whether **RES0** bits are stateful or **RAZ/WI**.

Type: `bool`

Default value: `false`

#### **`subcluster0.ete.RETSTACK`**

Return stack depth.

Type: `uint8_t`

Default value: 3

#### **`subcluster0.ete.REVISION`**

TRCIDR1 revision value.

Type: `uint8_t`

Default value: 0

#### **`subcluster0.ete.SIM_OVERFLOW_GRANULARITY`**

Number of instruction blocks in each granule, for simulated overflow.

Type: `uint32_t`

Default value: 0x64

#### **`subcluster0.ete.SIM_OVERFLOW_PERCENTAGE`**

Percentage of instruction blocks lost in each granule, for simulated overflow.

Type: `uint8_t`

Default value: 0

#### **`subcluster0.ete.SOURCE_ADDRESS`**

Allow generation of source address elements.

Type: `bool`

Default value: `false`

### **subcluster0.ete.TRACE\_OUTPUT**

File to which to write trace byte stream.

Type: `string`

Default value: `N/A`

### **subcluster0.ete.TRCSRSTA\_FORCED\_EXCEP**

TRCSRSTA value for a forcibly traced exception.

Type: `bool`

Default value: `false`

### **subcluster0.force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks**

Force MTE tag accesses to **RAZ/WI** and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

Type: `bool`

Default value: `false`

### **subcluster0.force\_zero\_PSTATE\_PAN**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

### **subcluster0.force\_zero\_mpam\_partid\_and\_pmg**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAMO\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

Type: `uint8_t`

Default value: `0`

### **subcluster0.has\_coherent\_icache**

Whether icache invalidation to the point of unification is required for instruction to data coherence.  
true - Invalidate operations not required.

Type: `bool`

Default value: `true`

#### **`subcluster0.has_delayed_dbgreg`**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

#### **`subcluster0.has_delayed_sysreg`**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

#### **`subcluster0.has_dot_product`**

Possible values of this parameter are:- 1, feature not implemented.- 2, feature is implemented.

Type: `uint8_t`

Default value: 2

#### **`subcluster0.has_ete`**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (`-plugin` or `-P`).

Type: `bool`

Default value: `false`

#### **`subcluster0.icache-hit_latency`**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.icache-maintenance_latency`**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

**subcluster0.icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.icache-size**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: `0x10000`

**subcluster0.invalidate\_code\_cache\_on\_icache\_cmo**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: `uint8_t`

Default value: 0

#### **`subcluster0.max_32bit_el`**

Maximum exception level supporting AArch32 modes. -1: No Support for A32 at any EL, 0: EL0 supports A32. This parameter is ignored in Rev0 i.e. when `revision_number` = 0.

Type: `int8_t`

Default value: 0

#### **`subcluster0.memory_tagging_support_level`**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

Type: `uint8_t`

Default value: 3

#### **`subcluster0.ptw_latency`**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

#### **`subcluster0.ras_pfg_clock_mhz`**

RAS Pseudo-Fault generation clock rate in MHz.

Type: `uint8_t`

Default value: 12

#### **`subcluster0.revision_number`**

Revision number of TRM implemented by the model. Changing the value can change the model behaviour. Visible as the Variant field in MIDR/MIDR\_EL1. Corresponds to the revision number X in rXpY.

Type: `uint8_t`

Default value: 1

#### **`subcluster0.tlb_latency`**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`



Default value: 0x0

#### **subcluster0.tlbi\_stall\_enabled**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: bool

Default value: false

#### **subcluster0.treat\_PAC\_as\_NOP**

Non-architecture parameter. Treat Pointer Authentication as **NOP**. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are **NOP**. Branch, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

Type: bool

Default value: false

#### **subcluster0.walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.NUM\_CORES**

Number of cores in the subcluster. Total number of cores in cluster may not exceed 12.

Type: uint8\_t

Default value: 1

#### **subcluster1.cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: uint32\_t

Default value: 0x1

#### **subcluster1.cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: uint32\_t

Default value: 0x1

**subcluster1.cpu0.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu0.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu0.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu0.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu0.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu0.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu0.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu0.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu0.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu0.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster1.cpu0.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu0.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu0.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu0.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu0.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu0.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu0.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu0.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu0.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu0.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster1.cpu0.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster1.cpu0.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu0.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu0.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu0.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu0.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu0.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu0.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu0.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu0.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu1.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu1.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu1.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu1.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

### **`subcluster1.cpu1.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`



**subcluster1.cpu1.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu1.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu1.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu1.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu1.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu1.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu1.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

### **`subcluster1.cpu1.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

### **`subcluster1.cpu1.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **`subcluster1.cpu1.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **`subcluster1.cpu1.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster1.cpu1.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster1.cpu1.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster1.cpu1.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster1.cpu1.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu1.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu10.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster1.cpu10.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu10.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu10.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter trace\_special\_hlt\_imm16.

Type: `bool`

Default value: `false`

**subcluster1.cpu10.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu10.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu10.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu10.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu10.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: 0x0

### **subcluster1.cpu10.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

### **subcluster1.cpu10.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster1.cpu10.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster1.cpu10.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster1.cpu10.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster1.cpu10.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu10.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu10.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu10.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu10.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu10.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster1.cpu10.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster1.cpu10.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster1.cpu10.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster1.cpu10.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu10.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu10.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu10.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu10.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu10.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster1.cpu10.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu2.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu2.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu2.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu2.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu2.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`



**subcluster1.cpu2.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

**subcluster1.cpu2.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster1.cpu2.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu2.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu2.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu2.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu2.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu2.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu2.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu2.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu2.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster1.cpu2.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster1.cpu2.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster1.cpu2.semihosting-cwd**

Base directory for semihosting file access.

Type: string

Default value: N/A

**subcluster1.cpu2.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: `true`

#### **`subcluster1.cpu2.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu2.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

#### **`subcluster1.cpu2.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

#### **`subcluster1.cpu2.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

#### **`subcluster1.cpu2.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster1.cpu2.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu3.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu3.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu3.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu3.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster1.cpu3.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu3.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu3.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu3.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu3.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu3.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster1.cpu3.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster1.cpu3.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu3.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu3.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu3.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`



**subcluster1.cpu3.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu3.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu3.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu3.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu3.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu4.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu4.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu4.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu4.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu4.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu4.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu4.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu4.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu4.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu4.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

#### **`subcluster1.cpu4.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu4.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu4.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu4.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu4.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu4.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu4.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu4.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu4.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu4.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**`subcluster1.cpu4.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**`subcluster1.cpu4.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**`subcluster1.cpu4.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**`subcluster1.cpu4.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**`subcluster1.cpu4.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**`subcluster1.cpu4.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**`subcluster1.cpu4.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

#### **`subcluster1.cpu4.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

#### **`subcluster1.cpu4.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster1.cpu4.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu5.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu5.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu5.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu5.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter trace\_special\_hlt\_imm16.

Type: `bool`

Default value: `false`

**subcluster1.cpu5.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu5.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu5.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu5.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu5.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: 0x0

### **subcluster1.cpu5.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

### **subcluster1.cpu5.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster1.cpu5.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster1.cpu5.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster1.cpu5.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster1.cpu5.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t



Default value: 0x100

**subcluster1.cpu5.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu5.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu5.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu5.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu5.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster1.cpu5.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster1.cpu5.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster1.cpu5.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster1.cpu5.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu5.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu5.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu5.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu5.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu5.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster1.cpu5.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu6.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu6.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu6.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu6.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu6.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu6.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu6.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu6.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu6.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu6.l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

**subcluster1.cpu6.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster1.cpu6.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu6.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu6.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu6.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu6.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu6.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu6.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu6.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu6.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster1.cpu6.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster1.cpu6.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster1.cpu6.semihosting-cwd**

Base directory for semihosting file access.

Type: string

Default value: N/A

**subcluster1.cpu6.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: `true`

### **`subcluster1.cpu6.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu6.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster1.cpu6.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster1.cpu6.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster1.cpu6.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu6.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu7.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu7.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu7.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu7.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu7.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu7.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`



**subcluster1.cpu7.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu7.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu7.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu7.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster1.cpu7.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu7.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu7.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu7.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu7.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu7.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu7.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu7.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu7.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu7.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster1.cpu7.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster1.cpu7.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu7.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu7.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu7.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu7.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu7.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu7.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu7.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu7.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu8.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu8.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu8.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu8.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu8.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu8.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu8.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu8.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu8.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu8.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

### **`subcluster1.cpu8.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu8.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu8.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu8.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu8.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu8.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu8.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu8.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu8.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu8.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**`subcluster1.cpu8.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**`subcluster1.cpu8.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**`subcluster1.cpu8.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**`subcluster1.cpu8.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**`subcluster1.cpu8.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**`subcluster1.cpu8.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**`subcluster1.cpu8.semihosting-stack_base`**

Virtual address of base of descending stack.



Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster1.cpu8.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster1.cpu8.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu8.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu9.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster1.cpu9.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu9.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu9.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter trace\_special\_hlt\_imm16.

Type: bool

Default value: false

**subcluster1.cpu9.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu9.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

#### **subcluster1.cpu9.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu9.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu9.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu9.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu9.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu9.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu9.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu9.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu9.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu9.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster1.cpu9.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster1.cpu9.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster1.cpu9.semihosting-cwd**

Base directory for semihosting file access.

Type: string

Default value: N/A

**subcluster1.cpu9.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: true

**subcluster1.cpu9.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu9.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster1.cpu9.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu9.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu9.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster1.dcache-hit_latency`**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.dcache-maintenance_latency`**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.dcache-miss_latency`**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.dcache-prefetch_enabled`**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

Type: `bool`

Default value: `false`

### **`subcluster1.dcache-read_access_latency`**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and

intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.dcache-read_latency`**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.dcache-size`**

L1 D-Cache size in bytes.

Type: `uint32_t`

Default value: `0x8000`

#### **`subcluster1.dcache-snoop_data_transfer_latency`**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.dcache-write_access_latency`**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.dcache-write_latency`**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster1.ete.CLAIMTAGS**

Number of claim tags.

Type: uint8\_t

Default value: 32

**subcluster1.ete.MAX\_INST\_PER\_Q**

Maximum limit for the number of instructions implied by a Q element.

Type: uint16\_t

Default value: 0x1

**subcluster1.ete.NumberOfRSPairs**

Number of resource selector pairs.

Type: uint8\_t

Default value: 8

**subcluster1.ete.PIDR\_CM0D**

TRCPIDR CM0D value.

Type: uint8\_t

Default value: 0

**subcluster1.ete.PIDR\_REVAND**

TRCPIDR REVAND value.

Type: uint8\_t

Default value: 1

**subcluster1.ete.PIDR\_REVISION**

TRCPIDR REVISION value.

Type: uint8\_t

Default value: 0

**subcluster1.ete.Q\_CADENCE**

Number of instruction blocks traced between two Q elements.

Type: uint16\_t



Default value: 0x1

**subcluster1.ete.RES0\_STATEFUL**

Whether **RES0** bits are stateful or **RAZ/WI**.

Type: bool

Default value: false

**subcluster1.ete.RETSTACK**

Return stack depth.

Type: uint8\_t

Default value: 3

**subcluster1.ete.REVISION**

TRCIDR1 revision value.

Type: uint8\_t

Default value: 2

**subcluster1.ete.SIM\_OVERFLOW\_GRANULARITY**

Number of instruction blocks in each granule, for simulated overflow.

Type: uint32\_t

Default value: 0x64

**subcluster1.ete.SIM\_OVERFLOW\_PERCENTAGE**

Percentage of instruction blocks lost in each granule, for simulated overflow.

Type: uint8\_t

Default value: 0

**subcluster1.ete.SOURCE\_ADDRESS**

Allow generation of source address elements.

Type: bool

Default value: false

**subcluster1.ete.TRACE\_OUTPUT**

File to which to write trace byte stream.

Type: string

Default value: `N/A`

**subcluster1.ete.TRCRSRTA\_FORCED\_EXCEP**

TRCRSR.TA value for a forcibly traced exception.

Type: `bool`

Default value: `false`

**subcluster1.ext\_abort\_device\_read\_is\_sync**

Synchronous reporting of device-nGnRE read external aborts.

Type: `bool`

Default value: `false`

**subcluster1.ext\_abort\_device\_write\_is\_sync**

Synchronous reporting of device-nGnRE write external aborts.

Type: `bool`

Default value: `false`

**subcluster1.ext\_abort\_so\_read\_is\_sync**

Synchronous reporting of device-nGnRnE read external aborts.

Type: `bool`

Default value: `false`

**subcluster1.ext\_abort\_so\_write\_is\_sync**

Synchronous reporting of device-nGnRnE write external aborts.

Type: `bool`

Default value: `false`

**subcluster1.force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks**

Force MTE tag accesses to **RAZ/WI** and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

Type: `bool`

Default value: `false`

**subcluster1.force\_zero\_PSTATE\_PAN**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

**subcluster1.force\_zero\_mpam\_partid\_and\_pmg**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

Type: `uint8_t`

Default value: `0`

**subcluster1.has\_coherent\_icache**

Whether icache invalidation to the point of unification is required for instruction to data coherence. `true` - Invalidate operations not required.

Type: `bool`

Default value: `true`

**subcluster1.has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**subcluster1.has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**subcluster1.has\_ete**

If `true`, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (`-plugin` or `-P`).

Type: `bool`

Default value: `false`

**subcluster1.icache-hit\_latency**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

Type: bool

Default value: false

**subcluster1.icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to

the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.icache-size`**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: `0x8000`

#### **`subcluster1.invalidate_code_cache_on_icache_cmo`**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: `uint8_t`

Default value: `0`

#### **`subcluster1.memory_tagging_support_level`**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).

Type: `uint8_t`

Default value: `2`

#### **`subcluster1.ptw_latency`**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.tlb_latency`**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.tlbi_stall_enabled`**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: `bool`

Default value: `false`

### **`subcluster1.treat-dcache-cmos-to-pou-as-nop`**

Whether dcache invalidation to the point of unification is required for instruction to data coherence. 0 - Invalidate ops required, 1 - Invalidate ops not required and cannot generate faults, 2 - Invalidate ops not required but can generate faults.

Type: `uint8_t`

Default value: 0

### **`subcluster1.walk_cache_latency`**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

## 3.46 ARMCortexA510CT\_CortexA710CT\_CortexX2CT

Defined in `LISA/ARMCortexA510CT_CortexA710CT_CortexX2CT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
CortexA510 r0p0	Full support
CortexA510 r1p3	Full support
CortexA710 r2p0	Full support
CortexX2 r2p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following parameters were added:

- `num_acp`
- `subcluster0.has_delayed_dbgreg`
- `subcluster0.has_delayed_sysreg`
- `subcluster1.has_delayed_dbgreg`
- `subcluster1.has_delayed_sysreg`
- `subcluster2.has_delayed_dbgreg`
- `subcluster2.has_delayed_sysreg`

## About ARMCortexA510CT\_CortexA710CT\_CortexX2CT

The number of cores in each subcluster is configurable using the following parameters:

### **subcluster0.NUM\_CORES**

Possible values are 1-10 (ARMCortexA510CT).

### **subcluster1.NUM\_CORES**

Possible values are 1-10 (ARMCortexA710CT).

### **subcluster2.NUM\_CORES**

Possible values are 1-10 (ARMCortexX2CT).

The total number of cores in the cluster cannot exceed 12.

Port arrays are expanded to the width needed for the maximum number of cores in each subcluster. Use the following port array indexes:

- <port\_name>[0-9] for cores in subcluster0.
- <port\_name>[10-19] for cores in subcluster1.
- <port\_name>[20-29] for cores in subcluster2.



#### Note

All instances in the Master cross trigger matrix port array, `cti[30]` must be connected, regardless of the `NUM_CORES` value used.

Core-specific parameters have the following prefixes:

- `subcluster0.cpu0` to `subcluster0.cpu9` identify cores in subcluster0.
- `subcluster1.cpu0` to `subcluster1.cpu9` identify cores in subcluster1.
- `subcluster2.cpu0` to `subcluster2.cpu9` identify cores in subcluster2.

For information about the cores in this model, see:

- [ARMCortexA510CT](#).
- [ARMCortexA710CT](#).
- [ARMCortexX2CT](#).

## Iris and MTI instances for ARMCortexA510CT\_CortexA710CT\_CortexX2CT

This model has the following Iris instances:

Name	Instance type
ARMCortexA510CT_CortexA710CT_CortexX2CT	<a href="#">Cluster_ARM_CortexA510_CortexA710_CortexX2_Heterogeneous</a>
ARMCortexA510CT_CortexA710CT_CortexX2CT.AMU	<a href="#">PVBUSLogger</a>
ARMCortexA510CT_CortexA710CT_CortexX2CT.AMU.mapper	<a href="#">PVBUSMapper</a>
ARMCortexA510CT_CortexA710CT_CortexX2CT.DAP	<a href="#">PVBUSLogger</a>

Name	Instance type
ARMCortexA510CT_CortexA710CT_CortexX2CT.DAP.mapper	PVBusMapper
ARMCortexA510CT_CortexA710CT_CortexX2CT.DSU	DSU-110
ARMCortexA510CT_CortexA710CT_CortexX2CT.DSU.PPU_cluster	PPUv1
ARMCortexA510CT_CortexA710CT_CortexX2CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.DSU.PPU_coreU (where $U = 0-2$ )	PPUv1
ARMCortexA510CT_CortexA710CT_CortexX2CT.DSU.PPU_coreU.busslave (where $U = 0-2$ )	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.DSU.shared_cache	PVCache
ARMCortexA510CT_CortexA710CT_CortexX2CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA510CT_CortexA710CT_CortexX2CT.DSU.shared_cache.upstream[U] (where $U = 0-7$ )	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.MMAP	PVBusLogger
ARMCortexA510CT_CortexA710CT_CortexX2CT.MMAP.mapper	PVBusMapper
ARMCortexA510CT_CortexA710CT_CortexX2CT.RAS	PVBusLogger
ARMCortexA510CT_CortexA710CT_CortexX2CT.RAS.mapper	PVBusMapper
ARMCortexA510CT_CortexA710CT_CortexX2CT.cpuU.debug_rom (where $U = 0-2$ )	debug_rom
ARMCortexA510CT_CortexA710CT_CortexX2CT.ext_bus	PVBusLogger
ARMCortexA510CT_CortexA710CT_CortexX2CT.ext_bus.mapper	PVBusMapper
ARMCortexA510CT_CortexA710CT_CortexX2CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA510CT_CortexA710CT_CortexX2CT.global_debug_rom	debug_rom
ARMCortexA510CT_CortexA710CT_CortexX2CT.secondary_debug_rom	debug_rom
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster0	Subcluster_ARM_Cortex-A510
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster0.cpu0	ARM_Cortex-A510
ARMCortexA510CT_CortexA710CT_CortexX2CT.subclusterU.cpu0.UTLB (where $U = 0-2$ )	TLB
ARMCortexA510CT_CortexA710CT_CortexX2CT.subclusterU.cpuV.dtlb (where $U = 0-2$ ; $V = 0-2$ )	TLB



Name	Instance type
ARMCortexA510CT_CortexA710CT_CortexX2CT.subclusterU.cpu0.l1dcache (where $U = 0-2$ )	PVCache
ARMCortexA510CT_CortexA710CT_CortexX2CT.subclusterU.cpu0.l1dcache.downstream[0].pvbusmaster (where $U = 0-2$ )	PVBusMaster
ARMCortexA510CT_CortexA710CT_CortexX2CT.subclusterU.cpu0.l1dcache.upstream[0] (where $U = 0-2$ )	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.subclusterU.cpu0.l1icache (where $U = 0-2$ )	PVCache
ARMCortexA510CT_CortexA710CT_CortexX2CT.subclusterU.cpu0.l1icache.downstream[0].pvbusmaster (where $U = 0-2$ )	PVBusMaster
ARMCortexA510CT_CortexA710CT_CortexX2CT.subclusterU.cpu0.l1icache.upstream[0] (where $U = 0-2$ )	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.subclusterU.cpu0.l2cache (where $U = 0-2$ )	PVCache
ARMCortexA510CT_CortexA710CT_CortexX2CT.subclusterU.cpu0.l2cache.downstream[0].pvbusmaster (where $U = 0-2$ )	PVBusMaster
ARMCortexA510CT_CortexA710CT_CortexX2CT.subclusterU.cpu0.l2cache.upstream[A] (where $U = 0-2$ ; $A = 0-1$ )	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.subclusterU.sve (where $U = 0-2$ )	ScalableVectorExtension
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster1	Subcluster_ARM_Cortex-A710
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster1.cpu0	ARM_Cortex-A710
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster2	Subcluster_ARM_Cortex-X2
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster2.cpu0	ARM_Cortex-X2

This model has the following MTI trace components:

Name	Component type
ARMCortexA510CT_CortexA710CT_CortexX2CT.AMU	PVBusLogger
ARMCortexA510CT_CortexA710CT_CortexX2CT.AMU.mapper	PVBusMapper
ARMCortexA510CT_CortexA710CT_CortexX2CT.DAP	PVBusLogger
ARMCortexA510CT_CortexA710CT_CortexX2CT.DAP.mapper	PVBusMapper
ARMCortexA510CT_CortexA710CT_CortexX2CT.DSU	DSU-110
ARMCortexA510CT_CortexA710CT_CortexX2CT.DSU.PPU_cluster	PPUv1
ARMCortexA510CT_CortexA710CT_CortexX2CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.DSU.PPU_coreU (where $U = 0-2$ )	PPUv1

Name	Component type
ARMCortexA510CT_CortexA710CT_CortexX2CT.DSU.PPU_coreU.busslave (where $U = 0-2$ )	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.DSU.shared_cache	PVCache
ARMCortexA510CT_CortexA710CT_CortexX2CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA510CT_CortexA710CT_CortexX2CT.DSU.shared_cache.upstream[U] (where $U = 0-7$ )	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.MMAP	PVBusLogger
ARMCortexA510CT_CortexA710CT_CortexX2CT.MMAP.mapper	PVBusMapper
ARMCortexA510CT_CortexA710CT_CortexX2CT.RAS	PVBusLogger
ARMCortexA510CT_CortexA710CT_CortexX2CT.RAS.mapper	PVBusMapper
ARMCortexA510CT_CortexA710CT_CortexX2CT.ext_bus	PVBusLogger
ARMCortexA510CT_CortexA710CT_CortexX2CT.ext_bus.mapper	PVBusMapper
ARMCortexA510CT_CortexA710CT_CortexX2CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster0.cpu0	ARM_Cortex-A510
ARMCortexA510CT_CortexA710CT_CortexX2CT.subclusterU.cpu0.UTLB (where $U = 0-2$ )	TLB
ARMCortexA510CT_CortexA710CT_CortexX2CT.subclusterU.cpu0.l1dcache (where $U = 0-2$ )	PVCache
ARMCortexA510CT_CortexA710CT_CortexX2CT.subclusterU.cpu0.l1dcache.downstream[0].pvbusmaster (where $U = 0-2$ )	PVBusMaster
ARMCortexA510CT_CortexA710CT_CortexX2CT.subclusterU.cpu0.l1dcache.upstream[0] (where $U = 0-2$ )	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.subclusterU.cpu0.l1icache (where $U = 0-2$ )	PVCache
ARMCortexA510CT_CortexA710CT_CortexX2CT.subclusterU.cpu0.l1icache.downstream[0].pvbusmaster (where $U = 0-2$ )	PVBusMaster
ARMCortexA510CT_CortexA710CT_CortexX2CT.subclusterU.cpu0.l1icache.upstream[0] (where $U = 0-2$ )	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.subclusterU.cpu0.l2cache (where $U = 0-2$ )	PVCache
ARMCortexA510CT_CortexA710CT_CortexX2CT.subclusterU.cpu0.l2cache.downstream[0].pvbusmaster (where $U = 0-2$ )	PVBusMaster
ARMCortexA510CT_CortexA710CT_CortexX2CT.subclusterU.cpu0.l2cache.upstream[A] (where $U = 0-2$ ; $A = 0-1$ )	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster1.cpu0	ARM_Cortex-A710
ARMCortexA510CT_CortexA710CT_CortexX2CT.subcluster2.cpu0	ARM_Cortex-X2

### Ports for ARMCortexA510CT\_CortexA710CT\_CortexX2CT

Port	Direction	Protocol	Description
acp_s	slave	PVBus	AXI ACP slave port.
AEND0MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).

Port	Direction	Protocol	Description
AEND2MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND3MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART2MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	slave	Signal	CHI defined pins.
broadcastcachemaint	slave	Signal	ACE defined pins.
broadcastouter	slave	Signal	ACE defined pins.
broadcastpersist	slave	Signal	CHI defined pins.
cfgend	slave	Signal	This signal is for EE bit initialisation.
cfgsdisable	slave	Signal	This signal disables write access to some secure Interrupt Controller registers.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcsmp_channel	master	PChannel	Cluster PCSM signal
cluster_powerdown_out	master	Signal	DEPRECATED - An output from PPU that informs thermal controller of the cluster power info
cluster_ppu_hw_stat	master	Value	Cluster PPU power state
clusterid	slave	Value	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set ManagerID64 of the core.
clusterpmuirq	master	Signal	DynamiQ pmu irq
CNTHPIRQ	master	Signal	Timer signals to SOC.
CNTHPSIRQ	master	Signal	Timer signals to SOC
CNTHVIRQ	master	Signal	Timer signals to SOC.
CNTHVSIRQ	master	Signal	Timer signals to SOC
CNTPNSIRQ	master	Signal	Timer signals to SOC.
CNTPSIRQ	master	Signal	Timer signals to SOC.
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.

Port	Direction	Protocol	Description
CNTVIRQ	master	Signal	Timer signals to SOC.
commirq	master	Signal	Interrupt signal from debug communications channel.
core_clk_in	slave	ClockSignal	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
core_pcsn_pchannel	master	PChannel	Core PCSM signals
core_powerdown_out	master	Signal	DEPRECATED - An output from PPUs that informs thermal controller of the core power info
core_ppu_hw_stat	master	Value	Notify the power state change inside the PPU.
coreerrirq	master	Signal	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq	master	Signal	Fault indicator for a detected 1 or 2 bit ECC error
coreinstrrun	master	Signal	Core Running State
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
ctidbgirq	master	Signal	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	slave	Signal	External debug interface.
dbgnopwrdown	master	Signal	No power-down request.
dbgprupreq	master	Signal	Debug power up request.
defaultMP	slave	Signal	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main master interface ports.
dev_debug_s	slave	PVBus	External debug interface.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.
gicreset	master	Signal	An output from PPUs that can be used to reset GICCLK domain components
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	slave	PVBus	External debug interface.
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	master	Signal	PPU cluster interrupt.
ppu_cluster_isolate	master	Signal	PPU Cluster Domain isolation control.
ppu_cluster_wakerequest	slave	Signal	PPU cluster wake request signal.
ppu_core_irq	master	Signal	PPU core interrupt.
ppu_core_isolate	master	Signal	PPU Core Domain isolation control.
ppu_core_wakerequest	slave	Signal	PPU Core wake request signals.
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	master	PVBus	The core will generate bus requests on this port.

Port	Direction	Protocol	Description
pvbus_periph_m	master	PVBus	The core can generate peripheral bus request on this port.
rei	slave	Signal	Per core RAM Error Interrupt.
reset	slave	Signal	Reset signal to cluster.
rvbaraddr	slave	Value_64	Reset vector base address.
sci_m	master	SystemCoherencyInterface	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei	slave	Signal	Per core System Error physical pins.
spiden	slave	Signal	External debug interface.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq	master	Signal	Interrupt signal from the trace buffer unit.
utility_bus_s	slave	PVBus	Utility bus slave
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfiq	slave	Signal	Virtualised FIQ.
virq	slave	Signal	Virtualised IRQ.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Per core virtual System Error physical pins.

## Parameters for ARM CortexA510CT\_CortexA710CT\_CortexX2CT

### AEND0\_DEFAULT

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

Type: uint64\_t

Default value: 0x0

### AEND1\_DEFAULT

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

Type: uint64\_t

Default value: 0x0

### AEND2\_DEFAULT

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

Type: uint64\_t

Default value: 0x0

### **AEND3\_DEFAULT**

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

Type: uint64\_t

Default value: 0x0

### **ASTART0\_DEFAULT**

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

Type: uint64\_t

Default value: 0x0

### **ASTART1\_DEFAULT**

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

Type: uint64\_t

Default value: 0x0

### **ASTART2\_DEFAULT**

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

Type: uint64\_t

Default value: 0x0

### **ASTART3\_DEFAULT**

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

Type: uint64\_t

Default value: 0x0

### **BROADCASTATOMIC**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: bool

Default value: `true`

### **BROADCASTCACHEMAINT**

Enable broadcasting of cache maintenance operations to downstream caches. The `broadcastcachemaint` signal will override this value if used.

Type: `bool`

Default value: `true`

### **BROADCASTOUTER**

Enable broadcasting of Outer Shareable transactions. The `broadcastouter` signal will override this value if used.

Type: `bool`

Default value: `true`

### **BROADCASTPERSIST**

Enable broadcasting of cache clean to the point of persistence operations. The `broadcastpersist` signal will override this value if used.

Type: `bool`

Default value: `true`

### **CLUSTER\_ID**

The Cluster ID maps to the affinity levels of the `MPIDR_EL1` register and is evaluated based on the `MPIDR_EL1` layout. If `MPIDR_EL1` supports 16-bit cluster affinity levels, bits [15:8] map to `IDRAFF3`, while bits [7:0] map to `IDRAFF2`. If `MPIDR_EL1` supports 24-bit cluster affinity levels, the bits [23:16] map to `IDRAFF3`, bits [15:8] map to `IDRAFF2`, and bits [7:0] map to `IDRAFF1`. This configuration also updates all relevant component `DEVAFF` registers and is used to set the `ManagerID64` of the core.

Type: `uint32_t`

Default value: `0x0`

### **CMO\_broadcast\_when\_cache\_state\_modelling\_disabled**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. This value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain events even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required.

Type: `uint8_t`

Default value: 1

### **GICDISABLE**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`

Default value: `true`

### **core\_power\_on\_by\_default**

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPUs which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

Type: `bool`

Default value: `false`

### **dcache-state\_modelled**

Set whether D-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **default\_opmode**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

Type: `uint8_t`

Default value: 4

### **diagnostics**

Enable DynamIQ diagnostic messages.

Type: `bool`

Default value: `false`

### **enable\_simulation\_performance\_optimizations**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of `stage12_tlb_size` parameter to 1024).

Type: `bool`



Default value: `true`

**has\_acp**

If true, Accelerator Coherency Port is configured.

Type: `bool`

Default value: `false`

**has\_peripheral\_port**

If true, an additional AXI peripheral port is configured.

Type: `bool`

Default value: `false`

**icache-state\_modelled**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

**l3cache-has\_mpam**

L3 Cache has MPAM support.

Type: `bool`

Default value: `false`

**l3cache-hit\_latency**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-maintenance\_latency**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-miss\_latency**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-read\_access\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size `l3cache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-read\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l3cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-size**

L3 Cache size in bytes.

Type: `uint32_t`

Default value: `0x0`

**l3cache-snoop\_data\_transfer\_latency**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-snoop\_issue\_latency**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-write_access_latency`**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size `l3cache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-write_latency`**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l3cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`mpmm_accumulator_multiplier`**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator. value of `n` means the accumulator will use (`n * accumulator value`) to calculate the mpmm threshold (MPMM). is provided as a fast model workaround to handle cases where execution of quantum in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

Type: `uint8_t`

Default value: `1`

### **`num_acp`**

Number of ACP ports.

Type: `uint8_t`

Default value: `0`

### **`num_nodes`**

Number of transport nodes. Zero implies direct-connect configuration.

Type: `uint8_t`

Default value: `1`

**periph\_address\_end**

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

Type: uint64\_t

Default value: 0x0

**periph\_address\_start**

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

Type: uint64\_t

Default value: 0x0

**subcluster0.CPUCFR**

Value of CPU Configuration Register.

Type: uint64\_t

Default value: 0x0

**subcluster0.NUM\_CORES**

Number of cores in the subcluster. Total number of cores in cluster may not exceed 12.

Type: uint8\_t

Default value: 1

**subcluster0.cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: uint32\_t

Default value: 0x1

**subcluster0.cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: uint32\_t

Default value: 0x1

**subcluster0.cpu0.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: `false`

**subcluster0.cpu0.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster0.cpu0.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu0.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu0.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu0.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu0.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu0.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu0.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu0.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster0.cpu0.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

### **subcluster0.cpu0.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster0.cpu0.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

### **subcluster0.cpu0.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster0.cpu0.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

### **subcluster0.cpu0.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu0.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu0.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu0.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu0.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster0.cpu0.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster0.cpu0.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu0.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`



**subcluster0.cpu0.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu0.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu0.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu0.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu0.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu0.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu0.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu0.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu1.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu1.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu1.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu1.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu1.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu1.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu1.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu1.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu1.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: `uint32_t`

Default value: `0x10`

**subcluster0.cpu1.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu1.l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: 0x80000

**subcluster0.cpu1.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu1.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu1.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu1.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu1.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu1.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu1.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu1.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster0.cpu1.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

### **`subcluster0.cpu1.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **`subcluster0.cpu1.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **`subcluster0.cpu1.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster0.cpu1.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu1.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu1.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster0.cpu1.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

#### **`subcluster0.cpu1.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster0.cpu1.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu1.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster0.cpu2.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu2.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu2.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: 0x0

**subcluster0.cpu2.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter trace\_special\_hlt\_imm16.

Type: bool

Default value: false

**subcluster0.cpu2.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t



Default value: 0x10

#### **subcluster0.cpu2.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu2.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

#### **subcluster0.cpu2.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu2.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu2.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

#### **subcluster0.cpu2.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu2.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu2.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu2.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu2.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu2.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu2.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster0.cpu2.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster0.cpu2.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu2.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu2.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu2.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu2.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu2.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu2.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu2.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu2.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster0.cpu2.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster0.cpu3.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster0.cpu3.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster0.cpu3.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu3.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu3.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu3.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu3.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu3.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu3.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu3.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu3.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu3.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu3.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu3.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu3.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu3.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster0.cpu3.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster0.cpu3.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string



Default value: N/A

**subcluster0.cpu3.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster0.cpu3.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu3.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu3.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu3.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu3.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu3.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster0.cpu3.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu3.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster0.cpu4.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu4.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu4.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu4.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu4.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu4.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu4.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

#### **subcluster0.cpu4.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu4.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu4.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

#### **subcluster0.cpu4.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu4.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu4.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu4.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu4.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu4.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu4.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu4.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

### **`subcluster0.cpu4.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

### **`subcluster0.cpu4.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **`subcluster0.cpu4.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **`subcluster0.cpu4.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster0.cpu4.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu4.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu4.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

#### **`subcluster0.cpu4.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

#### **`subcluster0.cpu4.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster0.cpu4.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu4.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster0.cpu5.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu5.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu5.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu5.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu5.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu5.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu5.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu5.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`



Default value: 0x0

**subcluster0.cpu5.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu5.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu5.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu5.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu5.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu5.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu5.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu5.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu5.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu5.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster0.cpu5.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster0.cpu5.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu5.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu5.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu5.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu5.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster0.cpu5.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu5.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu5.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster0.cpu5.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu6.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu6.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu6.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu6.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu6.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu6.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu6.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu6.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu6.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu6.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu6.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu6.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu6.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu6.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu6.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu6.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster0.cpu6.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster0.cpu6.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string



Default value: N/A

**subcluster0.cpu6.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster0.cpu6.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu6.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu6.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu6.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu6.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu6.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster0.cpu6.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu6.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster0.cpu7.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu7.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu7.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu7.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu7.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu7.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu7.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

#### **subcluster0.cpu7.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu7.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu7.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

#### **subcluster0.cpu7.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu7.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu7.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu7.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu7.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu7.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu7.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu7.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

### **`subcluster0.cpu7.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

### **`subcluster0.cpu7.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **`subcluster0.cpu7.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **`subcluster0.cpu7.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster0.cpu7.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu7.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu7.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster0.cpu7.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu7.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu7.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu7.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu8.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu8.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster0.cpu8.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu8.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu8.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu8.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu8.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu8.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`



Default value: 0x0

**subcluster0.cpu8.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu8.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu8.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu8.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu8.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu8.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu8.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu8.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu8.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu8.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster0.cpu8.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster0.cpu8.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu8.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu8.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu8.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu8.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu8.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu8.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu8.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu8.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster0.cpu8.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu9.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu9.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu9.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu9.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu9.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu9.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu9.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu9.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu9.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu9.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu9.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu9.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu9.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu9.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu9.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu9.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster0.cpu9.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster0.cpu9.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string



Default value: N/A

**subcluster0.cpu9.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster0.cpu9.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu9.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu9.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu9.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu9.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu9.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu9.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu9.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.dcache-hit_latency`**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.dcache-maintenance_latency`**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.dcache-miss_latency`**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.dcache-prefetch_enabled`**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

Type: `bool`

Default value: `false`

### **`subcluster0.dcache-read_access_latency`**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.dcache-read_latency`**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.dcache-size`**

L1 D-Cache size in bytes.

Type: `uint32_t`

Default value: `0x10000`

### **`subcluster0.dcache-snoop_data_transfer_latency`**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.dcache-write_access_latency`**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.ete.CLAIMTAGS**

Number of claim tags.

Type: uint8\_t

Default value: 4

**subcluster0.ete.MAX\_INST\_PER\_Q**

Maximum limit for the number of instructions implied by a Q element.

Type: uint16\_t

Default value: 0x1

**subcluster0.ete.NumberOfRSPairs**

Number of resource selector pairs.

Type: uint8\_t

Default value: 8

**subcluster0.ete.PIDR\_CM0D**

TRCPIDR CM0D value.

Type: uint8\_t

Default value: 0

**subcluster0.ete.PIDR\_REVAND**

TRCPIDR REVAND value.

Type: uint8\_t

Default value: 0

**subcluster0.ete.PIDR\_REVISION**

TRCPIDR REVISION value.

Type: `uint8_t`

Default value: 0

### **`subcluster0.ete.Q_CADENCE`**

Number of instruction blocks traced between two Q elements.

Type: `uint16_t`

Default value: 0x1

### **`subcluster0.ete.RES0_STATEFUL`**

Whether **RES0** bits are stateful or **RAZ/WI**.

Type: `bool`

Default value: `false`

### **`subcluster0.ete.RETSTACK`**

Return stack depth.

Type: `uint8_t`

Default value: 3

### **`subcluster0.ete.REVISION`**

TRCIDR1 revision value.

Type: `uint8_t`

Default value: 0

### **`subcluster0.ete.SIM_OVERFLOW_GRANULARITY`**

Number of instruction blocks in each granule, for simulated overflow.

Type: `uint32_t`

Default value: 0x64

### **`subcluster0.ete.SIM_OVERFLOW_PERCENTAGE`**

Percentage of instruction blocks lost in each granule, for simulated overflow.

Type: `uint8_t`

Default value: 0

### **`subcluster0.ete.SOURCE_ADDRESS`**

Allow generation of source address elements.

Type: `bool`

Default value: `false`

### **`subcluster0.ete.TRACE_OUTPUT`**

File to which to write trace byte stream.

Type: `string`

Default value: `N/A`

### **`subcluster0.ete.TRCSRSTA_FORCED_EXCEP`**

TRCSRSTA value for a forcibly traced exception.

Type: `bool`

Default value: `false`

### **`subcluster0.force_mte_tag_access_razwi_and_ignore_tag_checks`**

Force MTE tag accesses to **RAZ/WI** and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

Type: `bool`

Default value: `false`

### **`subcluster0.force_zero_PSTATE_PAN`**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

### **`subcluster0.force_zero_mpam_partid_and_pmg`**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAMO\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

Type: `uint8_t`

Default value: `0`

### **`subcluster0.has_coherent_icache`**

Whether icache invalidation to the point of unification is required for instruction to data coherence.  
true - Invalidate operations not required.

Type: `bool`

Default value: `true`

#### **`subcluster0.has_delayed_dbgreg`**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

#### **`subcluster0.has_delayed_sysreg`**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

#### **`subcluster0.has_dot_product`**

Possible values of this parameter are:- 1, feature not implemented.- 2, feature is implemented.

Type: `uint8_t`

Default value: 2

#### **`subcluster0.has_ete`**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (`-plugin` or `-P`).

Type: `bool`

Default value: `false`

#### **`subcluster0.icache-hit_latency`**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.icache-maintenance_latency`**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

**subcluster0.icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.icache-size**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: `0x10000`

**subcluster0.invalidate\_code\_cache\_on\_icache\_cmo**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.



Type: `uint8_t`

Default value: 0

#### **`subcluster0.max_32bit_el`**

Maximum exception level supporting AArch32 modes. -1: No Support for A32 at any EL, 0: EL0 supports A32. This parameter is ignored in Rev0 i.e. when `revision_number` = 0.

Type: `int8_t`

Default value: 0

#### **`subcluster0.memory_tagging_support_level`**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

Type: `uint8_t`

Default value: 3

#### **`subcluster0.ptw_latency`**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

#### **`subcluster0.ras_pfg_clock_mhz`**

RAS Pseudo-Fault generation clock rate in MHz.

Type: `uint8_t`

Default value: 12

#### **`subcluster0.revision_number`**

Revision number of TRM implemented by the model. Changing the value can change the model behaviour. Visible as the Variant field in MIDR/MIDR\_EL1. Corresponds to the revision number X in rXpY.

Type: `uint8_t`

Default value: 1

#### **`subcluster0.tlb_latency`**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

#### **subcluster0.tlbi\_stall\_enabled**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: bool

Default value: false

#### **subcluster0.treat\_PAC\_as\_NOP**

Non-architecture parameter. Treat Pointer Authentication as **NOP**. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are **NOP**. Branch, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

Type: bool

Default value: false

#### **subcluster0.walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.NUM\_CORES**

Number of cores in the subcluster. Total number of cores in cluster may not exceed 12.

Type: uint8\_t

Default value: 1

#### **subcluster1.cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: uint32\_t

Default value: 0x1

#### **subcluster1.cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: uint32\_t

Default value: 0x1

**subcluster1.cpu0.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu0.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu0.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu0.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu0.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu0.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu0.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu0.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu0.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu0.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster1.cpu0.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu0.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu0.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu0.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu0.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu0.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu0.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu0.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu0.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu0.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster1.cpu0.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster1.cpu0.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu0.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu0.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu0.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu0.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu0.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu0.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu0.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu0.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu1.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu1.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu1.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu1.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.



Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

### **`subcluster1.cpu1.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu1.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu1.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu1.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu1.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu1.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu1.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu1.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

### **`subcluster1.cpu1.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

### **`subcluster1.cpu1.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **`subcluster1.cpu1.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **`subcluster1.cpu1.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster1.cpu1.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster1.cpu1.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster1.cpu1.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster1.cpu1.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu1.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu2.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster1.cpu2.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu2.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter trace\_special\_hlt\_imm16.

Type: `bool`

Default value: `false`

**subcluster1.cpu2.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: 0x0

**subcluster1.cpu2.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster1.cpu2.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu2.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu2.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu2.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu2.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu2.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu2.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu2.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu2.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu2.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster1.cpu2.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster1.cpu2.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster1.cpu2.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster1.cpu2.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu2.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu2.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu2.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu2.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.



Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu2.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu3.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster1.cpu3.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu3.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu3.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu3.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

**subcluster1.cpu3.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster1.cpu3.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu3.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu3.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu3.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu3.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu3.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster1.cpu3.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster1.cpu3.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster1.cpu3.semihosting-cwd**

Base directory for semihosting file access.

Type: string

Default value: N/A

**subcluster1.cpu3.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: `true`

### **`subcluster1.cpu3.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu3.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster1.cpu3.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster1.cpu3.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster1.cpu3.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu3.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu4.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu4.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu4.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu4.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu4.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu4.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu4.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu4.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu4.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu4.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster1.cpu4.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu4.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu4.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu4.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu4.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu4.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu4.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu4.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000



**subcluster1.cpu4.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu4.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster1.cpu4.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster1.cpu4.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu4.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu4.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu4.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu4.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu4.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu4.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu4.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu4.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu5.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu5.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu5.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu5.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu5.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu5.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu5.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu5.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu5.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu5.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

#### **`subcluster1.cpu5.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu5.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu5.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu5.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu5.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu5.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu5.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu5.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu5.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu5.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

### **`subcluster1.cpu5.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

### **`subcluster1.cpu5.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **`subcluster1.cpu5.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **`subcluster1.cpu5.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster1.cpu5.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu5.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster1.cpu5.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster1.cpu5.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster1.cpu5.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu5.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu6.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster1.cpu6.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu6.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu6.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter trace\_special\_hlt\_imm16.

Type: `bool`

Default value: `false`

**subcluster1.cpu6.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu6.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu6.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu6.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu6.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`



Default value: 0x0

#### **subcluster1.cpu6.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

#### **subcluster1.cpu6.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu6.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu6.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu6.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu6.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu6.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu6.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu6.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu6.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu6.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster1.cpu6.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster1.cpu6.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster1.cpu6.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster1.cpu6.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu6.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu6.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu6.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu6.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu6.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster1.cpu6.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu7.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu7.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu7.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu7.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu7.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu7.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu7.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu7.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu7.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu7.l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

**subcluster1.cpu7.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster1.cpu7.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu7.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu7.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu7.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu7.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu7.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu7.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu7.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu7.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster1.cpu7.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster1.cpu7.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster1.cpu7.semihosting-cwd**

Base directory for semihosting file access.

Type: string

Default value: N/A

**subcluster1.cpu7.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: `true`

### **`subcluster1.cpu7.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu7.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster1.cpu7.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster1.cpu7.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster1.cpu7.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu7.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`



**subcluster1.cpu8.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu8.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu8.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu8.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu8.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu8.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu8.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu8.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu8.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu8.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster1.cpu8.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu8.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu8.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu8.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu8.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu8.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu8.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu8.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu8.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu8.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster1.cpu8.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster1.cpu8.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu8.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu8.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu8.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu8.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu8.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu8.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu8.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu8.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu9.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu9.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu9.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu9.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu9.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu9.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu9.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu9.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu9.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu9.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

### **`subcluster1.cpu9.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu9.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu9.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu9.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu9.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu9.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu9.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu9.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu9.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.



Type: `uint8_t`

Default value: 60

### **`subcluster1.cpu9.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

### **`subcluster1.cpu9.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **`subcluster1.cpu9.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **`subcluster1.cpu9.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster1.cpu9.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu9.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster1.cpu9.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster1.cpu9.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster1.cpu9.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu9.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster1.dcache-hit_latency`**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.dcache-maintenance_latency`**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.dcache-miss_latency`**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **subcluster1.dcache-prefetch\_enabled**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

Type: bool

Default value: false

### **subcluster1.dcache-read\_access\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster1.dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster1.dcache-size**

L1 D-Cache size in bytes.

Type: uint32\_t

Default value: 0x8000

### **subcluster1.dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster1.dcache-write\_access\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be

used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.dcache-write_latency`**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.ete.CLAIMTAGS`**

Number of claim tags.

Type: `uint8_t`

Default value: 32

#### **`subcluster1.ete.MAX_INST_PER_Q`**

Maximum limit for the number of instructions implied by a Q element.

Type: `uint16_t`

Default value: `0x1`

#### **`subcluster1.ete.NumberOfRSPairs`**

Number of resource selector pairs.

Type: `uint8_t`

Default value: 8

#### **`subcluster1.ete.PIDR_CM0D`**

TRCPIDR CM0D value.

Type: `uint8_t`

Default value: 0

#### **`subcluster1.ete.PIDR_REVAND`**

TRCPIDR REVAND value.

Type: `uint8_t`

Default value: 1

**subcluster1.ete.PIDR\_REVISION**

TRCPIDR REVISION value.

Type: `uint8_t`

Default value: 0

**subcluster1.ete.Q\_CADENCE**

Number of instruction blocks traced between two Q elements.

Type: `uint16_t`

Default value: 0x1

**subcluster1.ete.RES0\_STATEFUL**

Whether **RES0** bits are stateful or **RAZ/WI**.

Type: `bool`

Default value: `false`

**subcluster1.ete.RETSTACK**

Return stack depth.

Type: `uint8_t`

Default value: 3

**subcluster1.ete.REVISION**

TRCIDR1 revision value.

Type: `uint8_t`

Default value: 2

**subcluster1.ete.SIM\_OVERFLOW\_GRANULARITY**

Number of instruction blocks in each granule, for simulated overflow.

Type: `uint32_t`

Default value: 0x64

**subcluster1.ete.SIM\_OVERFLOW\_PERCENTAGE**

Percentage of instruction blocks lost in each granule, for simulated overflow.

Type: `uint8_t`

Default value: 0

**subcluster1.ete.SOURCE\_ADDRESS**

Allow generation of source address elements.

Type: bool

Default value: false

**subcluster1.ete.TRACE\_OUTPUT**

File to which to write trace byte stream.

Type: string

Default value: N/A

**subcluster1.ete.TRCRSRTA\_FORCED\_EXCEP**

TRCRSR.TA value for a forcibly traced exception.

Type: bool

Default value: false

**subcluster1.ext\_abort\_device\_read\_is\_sync**

Synchronous reporting of device-nGnRE read external aborts.

Type: bool

Default value: false

**subcluster1.ext\_abort\_device\_write\_is\_sync**

Synchronous reporting of device-nGnRE write external aborts.

Type: bool

Default value: false

**subcluster1.ext\_abort\_so\_read\_is\_sync**

Synchronous reporting of device-nGnRnE read external aborts.

Type: bool

Default value: false

**subcluster1.ext\_abort\_so\_write\_is\_sync**

Synchronous reporting of device-nGnRnE write external aborts.

Type: bool

Default value: `false`

### **subcluster1.force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks**

Force MTE tag accesses to **RAZ/WI** and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

Type: `bool`

Default value: `false`

### **subcluster1.force\_zero\_PSTATE\_PAN**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

### **subcluster1.force\_zero\_mpam\_partid\_and\_pmg**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAMO\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

Type: `uint8_t`

Default value: `0`

### **subcluster1.has\_coherent\_icache**

Whether icache invalidation to the point of unification is required for instruction to data coherence. true - Invalidate operations not required.

Type: `bool`

Default value: `true`

### **subcluster1.has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

### **subcluster1.has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

### **subcluster1.has\_ete**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (-plugin or -P).

Type: `bool`

Default value: `false`

### **subcluster1.icache-hit\_latency**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **subcluster1.icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **subcluster1.icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **subcluster1.icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

### **subcluster1.icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.



Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.icache-read_latency`**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.icache-size`**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: `0x8000`

### **`subcluster1.invalidate_code_cache_on_icache_cmo`**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: `uint8_t`

Default value: `0`

### **`subcluster1.memory_tagging_support_level`**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).

Type: `uint8_t`

Default value: `2`

### **`subcluster1.ptw_latency`**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.tlb_latency`**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

**subcluster1.tlbi\_stall\_enabled**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: bool

Default value: false

**subcluster1.treat-dcache-cmos-to-pou-as-nop**

Whether dcache invalidation to the point of unification is required for instruction to data coherence. 0 - Invalidate ops required, 1 - Invalidate ops not required and cannot generate faults, 2 - Invalidate ops not required but can generate faults.

Type: uint8\_t

Default value: 0

**subcluster1.walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

**subcluster2.NUM\_CORES**

Number of cores in the subcluster. Total number of cores in cluster may not exceed 12.

Type: uint8\_t

Default value: 1

**subcluster2.cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: uint32\_t

Default value: 0x1

**subcluster2.cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: uint32\_t

Default value: 0x1

**subcluster2.cpu0.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster2.cpu0.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster2.cpu0.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu0.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster2.cpu0.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu0.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu0.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu0.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu0.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu0.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster2.cpu0.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu0.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu0.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu0.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu0.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster2.cpu0.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster2.cpu0.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu0.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu0.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster2.cpu0.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster2.cpu0.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster2.cpu0.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster2.cpu0.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster2.cpu0.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster2.cpu0.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu0.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu0.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster2.cpu0.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu0.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster2.cpu0.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster2.cpu1.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster2.cpu1.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu1.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu1.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu1.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu1.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu1.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu1.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.



Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu1.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu1.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

### **`subcluster2.cpu1.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu1.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu1.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu1.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu1.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster2.cpu1.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster2.cpu1.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu1.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu1.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster2.cpu1.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

### **`subcluster2.cpu1.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

### **`subcluster2.cpu1.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **`subcluster2.cpu1.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **`subcluster2.cpu1.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster2.cpu1.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu1.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster2.cpu1.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster2.cpu1.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster2.cpu1.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster2.cpu1.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu2.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster2.cpu2.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu2.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu2.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter trace\_special\_hlt\_imm16.

Type: `bool`

Default value: `false`

**subcluster2.cpu2.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu2.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu2.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu2.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu2.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: 0x0

### **subcluster2.cpu2.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

### **subcluster2.cpu2.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster2.cpu2.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster2.cpu2.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster2.cpu2.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster2.cpu2.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster2.cpu2.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster2.cpu2.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu2.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu2.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster2.cpu2.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster2.cpu2.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster2.cpu2.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster2.cpu2.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster2.cpu2.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster2.cpu2.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu2.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu2.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster2.cpu2.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu2.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.



Type: `uint16_t`

Default value: `0xf000`

### **`subcluster2.cpu2.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu3.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster2.cpu3.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu3.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu3.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu3.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu3.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu3.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu3.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu3.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu3.l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

**subcluster2.cpu3.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster2.cpu3.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu3.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu3.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu3.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster2.cpu3.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster2.cpu3.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu3.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu3.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster2.cpu3.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster2.cpu3.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster2.cpu3.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster2.cpu3.semihosting-cwd**

Base directory for semihosting file access.

Type: string

Default value: N/A

**subcluster2.cpu3.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: `true`

### **`subcluster2.cpu3.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu3.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster2.cpu3.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster2.cpu3.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster2.cpu3.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster2.cpu3.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster2.cpu4.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster2.cpu4.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster2.cpu4.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu4.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster2.cpu4.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu4.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu4.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu4.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu4.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu4.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster2.cpu4.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu4.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster2.cpu4.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster2.cpu4.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster2.cpu4.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

#### **subcluster2.cpu4.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

#### **subcluster2.cpu4.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

#### **subcluster2.cpu4.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000



**subcluster2.cpu4.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster2.cpu4.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster2.cpu4.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster2.cpu4.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster2.cpu4.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster2.cpu4.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster2.cpu4.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu4.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu4.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster2.cpu4.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu4.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster2.cpu4.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster2.cpu5.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster2.cpu5.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **subcluster2.cpu5.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **subcluster2.cpu5.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **subcluster2.cpu5.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **subcluster2.cpu5.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **subcluster2.cpu5.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **subcluster2.cpu5.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu5.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu5.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

### **`subcluster2.cpu5.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu5.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu5.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu5.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu5.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster2.cpu5.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster2.cpu5.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu5.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu5.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster2.cpu5.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

### **`subcluster2.cpu5.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

### **`subcluster2.cpu5.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **`subcluster2.cpu5.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **`subcluster2.cpu5.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster2.cpu5.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu5.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster2.cpu5.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster2.cpu5.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster2.cpu5.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster2.cpu5.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu6.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster2.cpu6.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu6.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu6.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter trace\_special\_hlt\_imm16.

Type: `bool`

Default value: `false`

**subcluster2.cpu6.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu6.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu6.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu6.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu6.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`



Default value: 0x0

### **subcluster2.cpu6.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

### **subcluster2.cpu6.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster2.cpu6.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster2.cpu6.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster2.cpu6.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster2.cpu6.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster2.cpu6.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster2.cpu6.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu6.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu6.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster2.cpu6.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster2.cpu6.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster2.cpu6.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster2.cpu6.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster2.cpu6.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster2.cpu6.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu6.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu6.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster2.cpu6.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu6.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster2.cpu6.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu7.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster2.cpu7.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu7.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu7.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu7.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu7.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu7.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu7.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu7.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu7.l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

**subcluster2.cpu7.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

#### **subcluster2.cpu7.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster2.cpu7.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster2.cpu7.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster2.cpu7.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

#### **subcluster2.cpu7.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

#### **subcluster2.cpu7.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu7.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu7.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster2.cpu7.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster2.cpu7.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster2.cpu7.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster2.cpu7.semihosting-cwd**

Base directory for semihosting file access.

Type: string

Default value: N/A

**subcluster2.cpu7.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: `true`

### **`subcluster2.cpu7.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu7.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster2.cpu7.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster2.cpu7.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster2.cpu7.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster2.cpu7.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`



**subcluster2.cpu8.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster2.cpu8.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster2.cpu8.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu8.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster2.cpu8.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu8.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu8.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu8.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu8.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu8.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster2.cpu8.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu8.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu8.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu8.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu8.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster2.cpu8.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster2.cpu8.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu8.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu8.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster2.cpu8.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster2.cpu8.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster2.cpu8.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster2.cpu8.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster2.cpu8.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster2.cpu8.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu8.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu8.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster2.cpu8.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu8.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster2.cpu8.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster2.cpu9.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster2.cpu9.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu9.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu9.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu9.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu9.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu9.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu9.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu9.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu9.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

### **`subcluster2.cpu9.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu9.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu9.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu9.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu9.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster2.cpu9.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster2.cpu9.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu9.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu9.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster2.cpu9.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.



Type: `uint8_t`

Default value: 60

### **`subcluster2.cpu9.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

### **`subcluster2.cpu9.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **`subcluster2.cpu9.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **`subcluster2.cpu9.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster2.cpu9.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: 0x0

### **`subcluster2.cpu9.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: 0xf000000

### **`subcluster2.cpu9.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster2.cpu9.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster2.cpu9.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster2.cpu9.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster2.dcache-hit_latency`**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.dcache-maintenance_latency`**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.dcache-miss_latency`**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **subcluster2.dcache-prefetch\_enabled**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

Type: bool

Default value: false

### **subcluster2.dcache-read\_access\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster2.dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster2.dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster2.dcache-write\_access\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.etc.CLAIMTAGS**

Number of claim tags.

Type: uint8\_t

Default value: 32

**subcluster2.etc.MAX\_INST\_PER\_Q**

Maximum limit for the number of instructions implied by a Q element.

Type: uint16\_t

Default value: 0x1

**subcluster2.etc.NumberOfRSPairs**

Number of resource selector pairs.

Type: uint8\_t

Default value: 8

**subcluster2.etc.PIDR\_CM0D**

TRCPIDR CM0D value.

Type: uint8\_t

Default value: 0

**subcluster2.etc.PIDR\_REVAND**

TRCPIDR REVAND value.

Type: uint8\_t

Default value: 0

**subcluster2.etc.PIDR\_REVISION**

TRCPIDR REVISION value.

Type: `uint8_t`

Default value: 0

### **`subcluster2.ete.Q_CADENCE`**

Number of instruction blocks traced between two Q elements.

Type: `uint16_t`

Default value: 0x1

### **`subcluster2.ete.RES0_STATEFUL`**

Whether **RES0** bits are stateful or **RAZ/WI**.

Type: `bool`

Default value: `false`

### **`subcluster2.ete.RETSTACK`**

Return stack depth.

Type: `uint8_t`

Default value: 3

### **`subcluster2.ete.REVISION`**

TRCIDR1 revision value.

Type: `uint8_t`

Default value: 2

### **`subcluster2.ete.SIM_OVERFLOW_GRANULARITY`**

Number of instruction blocks in each granule, for simulated overflow.

Type: `uint32_t`

Default value: 0x64

### **`subcluster2.ete.SIM_OVERFLOW_PERCENTAGE`**

Percentage of instruction blocks lost in each granule, for simulated overflow.

Type: `uint8_t`

Default value: 0

### **`subcluster2.ete.SOURCE_ADDRESS`**

Allow generation of source address elements.

Type: `bool`

Default value: `false`

### **`subcluster2.ete.TRACE_OUTPUT`**

File to which to write trace byte stream.

Type: `string`

Default value: `N/A`

### **`subcluster2.ete.TRCSRSTA_FORCED_EXCEP`**

TRCSRSTA value for a forcibly traced exception.

Type: `bool`

Default value: `false`

### **`subcluster2.ext_abort_device_read_is_sync`**

Synchronous reporting of device-nGnRE read external aborts.

Type: `bool`

Default value: `false`

### **`subcluster2.ext_abort_device_write_is_sync`**

Synchronous reporting of device-nGnRE write external aborts.

Type: `bool`

Default value: `false`

### **`subcluster2.ext_abort_so_read_is_sync`**

Synchronous reporting of device-nGnRnE read external aborts.

Type: `bool`

Default value: `false`

### **`subcluster2.ext_abort_so_write_is_sync`**

Synchronous reporting of device-nGnRnE write external aborts.

Type: `bool`

Default value: `false`

**subcluster2.force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks**

Force MTE tag accesses to **RAZ/WI** and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

Type: `bool`

Default value: `false`

**subcluster2.force\_zero\_PSTATE\_PAN**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

**subcluster2.force\_zero\_mpam\_partid\_and\_pmg**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAMO\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

Type: `uint8_t`

Default value: `0`

**subcluster2.has\_coherent\_icache**

Whether icache invalidation to the point of unification is required for instruction to data coherence. true - Invalidate operations not required.

Type: `bool`

Default value: `true`

**subcluster2.has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**subcluster2.has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**subcluster2.has\_ete**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (-plugin or -P).

Type: `bool`

Default value: `false`

**subcluster2.icache-hit\_latency**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

**subcluster2.icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

Type: `uint64_t`



Default value: 0x0

### **subcluster2.icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **subcluster2.invalidate\_code\_cache\_on\_icache\_cmo**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: `uint8_t`

Default value: 0

### **subcluster2.memory\_tagging\_support\_level**

Specify the memory tagging extension support level: 0, not implemented. 1, instructions and registers only are implemented (FEAT\_MTE). 2, implemented (FEAT\_MTE2).

Type: `uint8_t`

Default value: 2

### **subcluster2.ptw\_latency**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

### **subcluster2.tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

### **subcluster2.tlbi\_stall\_enabled**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: `bool`

Default value: `false`

### **`subcluster2.treat-dcache-cmos-to-pou-as-nop`**

Whether dcache invalidation to the point of unification is required for instruction to data coherence. 0 - Invalidate ops required, 1 - Invalidate ops not required and cannot generate faults, 2 - Invalidate ops not required but can generate faults.

Type: `uint8_t`

Default value: 0

### **`subcluster2.walk_cache_latency`**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

## 3.47 ARMCortexA510CT\_CortexA710CT\_CortexX3CT

Defined in `LISA/ARMCortexA510CT_CortexA710CT_CortexX3CT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
CortexA510 r0p0	Full support
CortexA510 r1p3	Full support
CortexA710 r2p0	Full support
CortexX3 r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following parameters were added:

- `num_acp`
- `subcluster0.has_delayed_dbgreg`
- `subcluster0.has_delayed_sysreg`
- `subcluster1.has_delayed_dbgreg`
- `subcluster1.has_delayed_sysreg`
- `subcluster2.has_delayed_dbgreg`
- `subcluster2.has_delayed_sysreg`

## About ARMCortexA510CT\_CortexA710CT\_CortexX3CT

The number of cores in each subcluster is configurable using the following parameters:

### **subcluster0.NUM\_CORES**

Possible values are 1-10 (ARMCortexA510CT).

### **subcluster1.NUM\_CORES**

Possible values are 1-10 (ARMCortexA710CT).

### **subcluster2.NUM\_CORES**

Possible values are 1-10 (ARMCortexX3CT).

The total number of cores in the cluster cannot exceed 12.

Port arrays are expanded to the width needed for the maximum number of cores in each subcluster. Use the following port array indexes:

- <port\_name>[0-9] for cores in subcluster0.
- <port\_name>[10-19] for cores in subcluster1.
- <port\_name>[20-29] for cores in subcluster2.



#### Note

All instances in the Master cross trigger matrix port array, `cti[30]` must be connected, regardless of the `NUM_CORES` value used.

Core-specific parameters have the following prefixes:

- `subcluster0.cpu0` to `subcluster0.cpu9` identify cores in subcluster0.
- `subcluster1.cpu0` to `subcluster1.cpu9` identify cores in subcluster1.
- `subcluster2.cpu0` to `subcluster2.cpu9` identify cores in subcluster2.

For information about the cores in this model, see:

- [ARMCortexA510CT](#).
- [ARMCortexA710CT](#).
- [ARMCortexX3CT](#).

## Iris and MTI instances for ARMCortexA510CT\_CortexA710CT\_CortexX3CT

This model has the following Iris instances:

Name	Instance type
ARMCortexA510CT_CortexA710CT_CortexX3CT	<a href="#">Cluster_ARM_CortexA510_CortexA710_CortexX3_Heterogeneous</a>
ARMCortexA510CT_CortexA710CT_CortexX3CT.AMU	<a href="#">PVBusLogger</a>
ARMCortexA510CT_CortexA710CT_CortexX3CT.AMU.mapper	<a href="#">PVBusMapper</a>
ARMCortexA510CT_CortexA710CT_CortexX3CT.DAP	<a href="#">PVBusLogger</a>

Name	Instance type
ARMCortexA510CT_CortexA710CT_CortexX3CT.DAP.mapper	PVBusMapper
ARMCortexA510CT_CortexA710CT_CortexX3CT.DSU	DSU-110
ARMCortexA510CT_CortexA710CT_CortexX3CT.DSU.PPU_cluster	PPUv1
ARMCortexA510CT_CortexA710CT_CortexX3CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.DSU.PPU_coreU (where $U = 0-2$ )	PPUv1
ARMCortexA510CT_CortexA710CT_CortexX3CT.DSU.PPU_coreU.busslave (where $U = 0-2$ )	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.DSU.shared_cache	PVCache
ARMCortexA510CT_CortexA710CT_CortexX3CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA510CT_CortexA710CT_CortexX3CT.DSU.shared_cache.upstream[U] (where $U = 0-7$ )	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.MMAP	PVBusLogger
ARMCortexA510CT_CortexA710CT_CortexX3CT.MMAP.mapper	PVBusMapper
ARMCortexA510CT_CortexA710CT_CortexX3CT.RAS	PVBusLogger
ARMCortexA510CT_CortexA710CT_CortexX3CT.RAS.mapper	PVBusMapper
ARMCortexA510CT_CortexA710CT_CortexX3CT.cpuU.debug_rom (where $U = 0-2$ )	debug_rom
ARMCortexA510CT_CortexA710CT_CortexX3CT.ext_bus	PVBusLogger
ARMCortexA510CT_CortexA710CT_CortexX3CT.ext_bus.mapper	PVBusMapper
ARMCortexA510CT_CortexA710CT_CortexX3CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA510CT_CortexA710CT_CortexX3CT.global_debug_rom	debug_rom
ARMCortexA510CT_CortexA710CT_CortexX3CT.secondary_debug_rom	debug_rom
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster0	Subcluster_ARM_Cortex-A510
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster0.cpu0	ARM_Cortex-A510
ARMCortexA510CT_CortexA710CT_CortexX3CT.subclusterU.cpu0.UTLB (where $U = 0-2$ )	TLB
ARMCortexA510CT_CortexA710CT_CortexX3CT.subclusterU.cpuV.dtlb (where $U = 0-2$ ; $V = 0-2$ )	TLB

Name	Instance type
ARMCortexA510CT_CortexA710CT_CortexX3CT.subclusterU.cpu0.l1dcache (where $U = 0-2$ )	PVCache
ARMCortexA510CT_CortexA710CT_CortexX3CT.subclusterU.cpu0.l1dcache.downstream[0].pvbusmaster (where $U = 0-2$ )	PVBusMaster
ARMCortexA510CT_CortexA710CT_CortexX3CT.subclusterU.cpu0.l1dcache.upstream[0] (where $U = 0-2$ )	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.subclusterU.cpu0.l1icache (where $U = 0-2$ )	PVCache
ARMCortexA510CT_CortexA710CT_CortexX3CT.subclusterU.cpu0.l1icache.downstream[0].pvbusmaster (where $U = 0-2$ )	PVBusMaster
ARMCortexA510CT_CortexA710CT_CortexX3CT.subclusterU.cpu0.l1icache.upstream[0] (where $U = 0-2$ )	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.subclusterU.cpu0.l2cache (where $U = 0-2$ )	PVCache
ARMCortexA510CT_CortexA710CT_CortexX3CT.subclusterU.cpu0.l2cache.downstream[0].pvbusmaster (where $U = 0-2$ )	PVBusMaster
ARMCortexA510CT_CortexA710CT_CortexX3CT.subclusterU.cpu0.l2cache.upstream[A] (where $U = 0-2$ ; $A = 0-1$ )	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.subclusterU.sve (where $U = 0-2$ )	ScalableVectorExtension
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster1	Subcluster_ARM_Cortex-A710
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster1.cpu0	ARM_Cortex-A710
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster2	Subcluster_ARM_Cortex-X3
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster2.cpu0	ARM_Cortex-X3

This model has the following MTI trace components:

Name	Component type
ARMCortexA510CT_CortexA710CT_CortexX3CT.AMU	PVBusLogger
ARMCortexA510CT_CortexA710CT_CortexX3CT.AMU.mapper	PVBusMapper
ARMCortexA510CT_CortexA710CT_CortexX3CT.DAP	PVBusLogger
ARMCortexA510CT_CortexA710CT_CortexX3CT.DAP.mapper	PVBusMapper
ARMCortexA510CT_CortexA710CT_CortexX3CT.DSU	DSU-110
ARMCortexA510CT_CortexA710CT_CortexX3CT.DSU.PPU_cluster	PPUv1
ARMCortexA510CT_CortexA710CT_CortexX3CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.DSU.PPU_coreU (where $U = 0-2$ )	PPUv1

Name	Component type
ARMCortexA510CT_CortexA710CT_CortexX3CT.DSU.PPU_coreU.busslave (where $U = 0-2$ )	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.DSU.shared_cache	PVCache
ARMCortexA510CT_CortexA710CT_CortexX3CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA510CT_CortexA710CT_CortexX3CT.DSU.shared_cache.upstream[U] (where $U = 0-7$ )	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.MMAP	PVBusLogger
ARMCortexA510CT_CortexA710CT_CortexX3CT.MMAP.mapper	PVBusMapper
ARMCortexA510CT_CortexA710CT_CortexX3CT.RAS	PVBusLogger
ARMCortexA510CT_CortexA710CT_CortexX3CT.RAS.mapper	PVBusMapper
ARMCortexA510CT_CortexA710CT_CortexX3CT.ext_bus	PVBusLogger
ARMCortexA510CT_CortexA710CT_CortexX3CT.ext_bus.mapper	PVBusMapper
ARMCortexA510CT_CortexA710CT_CortexX3CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster0.cpu0	ARM_Cortex-A510
ARMCortexA510CT_CortexA710CT_CortexX3CT.subclusterU.cpu0.UTLB (where $U = 0-2$ )	TLB
ARMCortexA510CT_CortexA710CT_CortexX3CT.subclusterU.cpu0.l1dcache (where $U = 0-2$ )	PVCache
ARMCortexA510CT_CortexA710CT_CortexX3CT.subclusterU.cpu0.l1dcache.downstream[0].pvbusmaster (where $U = 0-2$ )	PVBusMaster
ARMCortexA510CT_CortexA710CT_CortexX3CT.subclusterU.cpu0.l1dcache.upstream[0] (where $U = 0-2$ )	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.subclusterU.cpu0.l1icache (where $U = 0-2$ )	PVCache
ARMCortexA510CT_CortexA710CT_CortexX3CT.subclusterU.cpu0.l1icache.downstream[0].pvbusmaster (where $U = 0-2$ )	PVBusMaster
ARMCortexA510CT_CortexA710CT_CortexX3CT.subclusterU.cpu0.l1icache.upstream[0] (where $U = 0-2$ )	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.subclusterU.cpu0.l2cache (where $U = 0-2$ )	PVCache
ARMCortexA510CT_CortexA710CT_CortexX3CT.subclusterU.cpu0.l2cache.downstream[0].pvbusmaster (where $U = 0-2$ )	PVBusMaster
ARMCortexA510CT_CortexA710CT_CortexX3CT.subclusterU.cpu0.l2cache.upstream[A] (where $U = 0-2$ ; $A = 0-1$ )	PVBusSlave
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster1.cpu0	ARM_Cortex-A710
ARMCortexA510CT_CortexA710CT_CortexX3CT.subcluster2.cpu0	ARM_Cortex-X3

### Ports for ARMCortexA510CT\_CortexA710CT\_CortexX3CT

Port	Direction	Protocol	Description
acp_s	slave	PVBus	AXI ACP slave port.
AEND0MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).

Port	Direction	Protocol	Description
AEND2MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND3MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART2MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	slave	Signal	CHI defined pins.
broadcastcachemaint	slave	Signal	ACE defined pins.
broadcastouter	slave	Signal	ACE defined pins.
broadcastpersist	slave	Signal	CHI defined pins.
cfgend	slave	Signal	This signal is for EE bit initialisation.
cfgsdisable	slave	Signal	This signal disables write access to some secure Interrupt Controller registers.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcsmp_channel	master	PChannel	Cluster PCSM signal
cluster_powerdown_out	master	Signal	DEPRECATED - An output from PPU that informs thermal controller of the cluster power info
cluster_ppu_hw_stat	master	Value	Cluster PPU power state
clusterid	slave	Value	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set ManagerID64 of the core.
clusterpmuirq	master	Signal	DynamiQ pmu irq
CNTHPIRQ	master	Signal	Timer signals to SOC.
CNTHPSIRQ	master	Signal	Timer signals to SOC
CNTHVIRQ	master	Signal	Timer signals to SOC.
CNTHVSIRQ	master	Signal	Timer signals to SOC
CNTPNSIRQ	master	Signal	Timer signals to SOC.
CNTPSIRQ	master	Signal	Timer signals to SOC.
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.

Port	Direction	Protocol	Description
CNTVIRQ	master	Signal	Timer signals to SOC.
commirq	master	Signal	Interrupt signal from debug communications channel.
core_clk_in	slave	ClockSignal	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
core_pcsn_pchannel	master	PChannel	Core PCSM signals
core_powerdown_out	master	Signal	DEPRECATED - An output from PPUs that informs thermal controller of the core power info
core_ppu_hw_stat	master	Value	Notify the power state change inside the PPU.
coreerrirq	master	Signal	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq	master	Signal	Fault indicator for a detected 1 or 2 bit ECC error
coreinstrrun	master	Signal	Core Running State
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
ctidbgirq	master	Signal	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	slave	Signal	External debug interface.
dbgnopwrdown	master	Signal	No power-down request.
dbgprupreq	master	Signal	Debug power up request.
defaultMP	slave	Signal	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main master interface ports.
dev_debug_s	slave	PVBus	External debug interface.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.
gicreset	master	Signal	An output from PPUs that can be used to reset GICCLK domain components
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	slave	PVBus	External debug interface.
pmbirq	master	Signal	Interrupt signal from the statistical profiling unit.
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	master	Signal	PPU cluster interrupt.
ppu_cluster_isolate	master	Signal	PPU Cluster Domain isolation control.
ppu_cluster_wakerequest	slave	Signal	PPU cluster wake request signal.
ppu_core_irq	master	Signal	PPU core interrupt.
ppu_core_isolate	master	Signal	PPU Core Domain isolation control.
ppu_core_wakerequest	slave	Signal	PPU Core wake request signals.
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.



Port	Direction	Protocol	Description
pvbus_m0	master	PVBus	The core will generate bus requests on this port.
pvbus_periph_m	master	PVBus	The core can generate peripheral bus request on this port.
rei	slave	Signal	Per core RAM Error Interrupt.
reset	slave	Signal	Reset signal to cluster.
rvbaraddr	slave	Value_64	Reset vector base address.
sci_m	master	SystemCoherencyInterface	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei	slave	Signal	Per core System Error physical pins.
spiden	slave	Signal	External debug interface.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq	master	Signal	Interrupt signal from the trace buffer unit.
utility_bus_s	slave	PVBus	Utility bus slave
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfiq	slave	Signal	Virtualised FIQ.
virq	slave	Signal	Virtualised IRQ.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Per core virtual System Error physical pins.

## Parameters for ARMCortexA510CT\_CortexA710CT\_CortexX3CT

### AEND0\_DEFAULT

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

Type: uint64\_t

Default value: 0x0

### AEND1\_DEFAULT

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

Type: uint64\_t

Default value: 0x0

### AEND2\_DEFAULT

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

Type: `uint64_t`

Default value: `0x0`

### **AEND3\_DEFAULT**

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

Type: `uint64_t`

Default value: `0x0`

### **ASTART0\_DEFAULT**

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

Type: `uint64_t`

Default value: `0x0`

### **ASTART1\_DEFAULT**

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

Type: `uint64_t`

Default value: `0x0`

### **ASTART2\_DEFAULT**

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

Type: `uint64_t`

Default value: `0x0`

### **ASTART3\_DEFAULT**

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

Type: `uint64_t`

Default value: `0x0`

### **BROADCASTATOMIC**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`

Default value: `true`

### **BROADCASTCACHEMAINT**

Enable broadcasting of cache maintenance operations to downstream caches. The `broadcastcachemaint` signal will override this value if used.

Type: `bool`

Default value: `true`

### **BROADCASTOUTER**

Enable broadcasting of Outer Shareable transactions. The `broadcastouter` signal will override this value if used.

Type: `bool`

Default value: `true`

### **BROADCASTPERSIST**

Enable broadcasting of cache clean to the point of persistence operations. The `broadcastpersist` signal will override this value if used.

Type: `bool`

Default value: `true`

### **CLUSTER\_ID**

The Cluster ID maps to the affinity levels of the `MPIDR_EL1` register and is evaluated based on the `MPIDR_EL1` layout. If `MPIDR_EL1` supports 16-bit cluster affinity levels, bits [15:8] map to `IDRAFF3`, while bits [7:0] map to `IDRAFF2`. If `MPIDR_EL1` supports 24-bit cluster affinity levels, the bits [23:16] map to `IDRAFF3`, bits [15:8] map to `IDRAFF2`, and bits [7:0] map to `IDRAFF1`. This configuration also updates all relevant component `DEVAFF` registers and is used to set the `ManagerID64` of the core.

Type: `uint32_t`

Default value: `0x0`

### **CMO\_broadcast\_when\_cache\_state\_modelling\_disabled**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. This value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required.

Type: `uint8_t`

Default value: 1

### **GICDISABLE**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`

Default value: `true`

### **core\_power\_on\_by\_default**

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPUs which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

Type: `bool`

Default value: `false`

### **dcache-state\_modelled**

Set whether D-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **default\_opmode**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

Type: `uint8_t`

Default value: 4

### **diagnostics**

Enable DynamIQ diagnostic messages.

Type: `bool`

Default value: `false`

### **enable\_simulation\_performance\_optimizations**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of `stage12_tlb_size` parameter to 1024).

Type: `bool`

Default value: `true`

### **has\_acp**

If true, Accelerator Coherency Port is configured.

Type: `bool`

Default value: `false`

### **has\_peripheral\_port**

If true, an additional AXI peripheral port is configured.

Type: `bool`

Default value: `false`

### **icache-state\_modelled**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **l3cache-has\_mpam**

L3 Cache has MPAM support.

Type: `bool`

Default value: `false`

### **l3cache-hit\_latency**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **l3cache-maintenance\_latency**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-miss\_latency**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-read\_access\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size `l3cache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-read\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l3cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-size**

L3 Cache size in bytes.

Type: `uint32_t`

Default value: `0x0`

**l3cache-snoop\_data\_transfer\_latency**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-snoop\_issue\_latency**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-write_access_latency`**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size `l3cache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-write_latency`**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l3cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`mpmm_accumulator_multiplier`**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator.value of `n` means the accumulator will use (`n * accumulator value`) to calculate the mpmm threshold (MPMM).is provided as a fast model workaround to handle cases where execution of quantum in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

Type: `uint8_t`

Default value: `1`

### **`num_acp`**

Number of ACP ports.

Type: `uint8_t`

Default value: `0`

### **`num_nodes`**

Number of transport nodes. Zero implies direct-connect configuration.

Type: `uint8_t`

Default value: `1`

**periph\_address\_end**

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

Type: uint64\_t

Default value: 0x0

**periph\_address\_start**

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

Type: uint64\_t

Default value: 0x0

**subcluster0.CPUCFR**

Value of CPU Configuration Register.

Type: uint64\_t

Default value: 0x0

**subcluster0.NUM\_CORES**

Number of cores in the subcluster. Total number of cores in cluster may not exceed 12.

Type: uint8\_t

Default value: 1

**subcluster0.cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: uint32\_t

Default value: 0x1

**subcluster0.cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: uint32\_t

Default value: 0x1

**subcluster0.cpu0.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: bool



Default value: `false`

#### **`subcluster0.cpu0.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu0.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu0.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu0.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu0.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu0.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu0.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu0.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu0.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster0.cpu0.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

### **subcluster0.cpu0.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster0.cpu0.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

### **subcluster0.cpu0.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster0.cpu0.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

### **subcluster0.cpu0.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu0.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu0.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu0.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu0.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster0.cpu0.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster0.cpu0.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu0.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu0.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu0.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu0.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu0.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu0.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu0.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu0.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu0.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu1.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu1.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu1.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu1.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu1.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu1.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu1.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu1.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu1.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: `uint32_t`

Default value: `0x10`

**subcluster0.cpu1.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu1.l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: 0x80000

**subcluster0.cpu1.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu1.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu1.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t



Default value: 0x0

**subcluster0.cpu1.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu1.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu1.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu1.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu1.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu1.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster0.cpu1.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

### **`subcluster0.cpu1.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **`subcluster0.cpu1.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **`subcluster0.cpu1.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster0.cpu1.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu1.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu1.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster0.cpu1.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

#### **`subcluster0.cpu1.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster0.cpu1.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu1.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster0.cpu2.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu2.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu2.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: 0x0

**subcluster0.cpu2.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter trace\_special\_hlt\_imm16.

Type: bool

Default value: false

**subcluster0.cpu2.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

### **subcluster0.cpu2.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster0.cpu2.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

### **subcluster0.cpu2.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster0.cpu2.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster0.cpu2.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

### **subcluster0.cpu2.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu2.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu2.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu2.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu2.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu2.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu2.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster0.cpu2.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster0.cpu2.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu2.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu2.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu2.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu2.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu2.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu2.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu2.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu2.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster0.cpu2.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`



**subcluster0.cpu3.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster0.cpu3.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster0.cpu3.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu3.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu3.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu3.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu3.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu3.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu3.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu3.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu3.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu3.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu3.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu3.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu3.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu3.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster0.cpu3.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster0.cpu3.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster0.cpu3.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster0.cpu3.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu3.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu3.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu3.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu3.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu3.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster0.cpu3.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu3.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster0.cpu4.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu4.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu4.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu4.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu4.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu4.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu4.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu4.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20



**subcluster0.cpu4.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu4.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu4.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu4.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu4.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu4.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

### **`subcluster0.cpu4.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

### **`subcluster0.cpu4.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **`subcluster0.cpu4.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **`subcluster0.cpu4.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster0.cpu4.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu4.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu4.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

#### **`subcluster0.cpu4.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

#### **`subcluster0.cpu4.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster0.cpu4.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu4.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster0.cpu5.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu5.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu5.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu5.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu5.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu5.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu5.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu5.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster0.cpu5.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu5.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu5.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu5.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu5.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu5.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu5.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu5.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu5.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu5.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster0.cpu5.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster0.cpu5.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu5.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu5.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu5.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu5.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster0.cpu5.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu5.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu5.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster0.cpu5.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).



Type: `bool`

Default value: `true`

### **subcluster0.cpu6.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **subcluster0.cpu6.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **subcluster0.cpu6.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **subcluster0.cpu6.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **subcluster0.cpu6.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **subcluster0.cpu6.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu6.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu6.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu6.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu6.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu6.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu6.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu6.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu6.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu6.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu6.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster0.cpu6.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster0.cpu6.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster0.cpu6.semihosting-cwd**

Base directory for semihosting file access.

Type: string

Default value: N/A

**subcluster0.cpu6.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: true

**subcluster0.cpu6.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu6.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster0.cpu6.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu6.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster0.cpu6.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu6.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster0.cpu7.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu7.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu7.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu7.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu7.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu7.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu7.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu7.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20



**subcluster0.cpu7.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu7.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu7.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu7.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu7.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu7.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**`subcluster0.cpu7.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**`subcluster0.cpu7.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**`subcluster0.cpu7.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**`subcluster0.cpu7.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**`subcluster0.cpu7.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**`subcluster0.cpu7.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**`subcluster0.cpu7.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster0.cpu7.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu7.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu7.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu7.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu8.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu8.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu8.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu8.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu8.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu8.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu8.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu8.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster0.cpu8.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu8.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu8.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu8.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu8.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu8.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu8.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu8.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu8.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu8.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster0.cpu8.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster0.cpu8.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu8.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu8.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu8.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu8.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu8.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu8.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu8.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu8.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster0.cpu8.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).



Type: `bool`

Default value: `true`

### **subcluster0.cpu9.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **subcluster0.cpu9.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **subcluster0.cpu9.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **subcluster0.cpu9.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **subcluster0.cpu9.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **subcluster0.cpu9.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu9.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu9.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu9.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu9.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu9.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu9.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu9.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu9.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu9.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu9.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster0.cpu9.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster0.cpu9.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster0.cpu9.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster0.cpu9.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu9.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu9.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu9.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu9.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu9.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster0.cpu9.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu9.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster0.dcache-hit_latency`**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.dcache-maintenance_latency`**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.dcache-miss_latency`**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.dcache-prefetch_enabled`**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

Type: `bool`

Default value: `false`

### **`subcluster0.dcache-read_access_latency`**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.dcache-read_latency`**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.dcache-size`**

L1 D-Cache size in bytes.

Type: `uint32_t`

Default value: `0x10000`

### **`subcluster0.dcache-snoop_data_transfer_latency`**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.dcache-write_access_latency`**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.ete.CLAIMTAGS**

Number of claim tags.

Type: uint8\_t

Default value: 4

**subcluster0.ete.MAX\_INST\_PER\_Q**

Maximum limit for the number of instructions implied by a Q element.

Type: uint16\_t

Default value: 0x1

**subcluster0.ete.NumberOfRSPairs**

Number of resource selector pairs.

Type: uint8\_t

Default value: 8

**subcluster0.ete.PIDR\_CM0D**

TRCPIDR CM0D value.

Type: uint8\_t

Default value: 0

**subcluster0.ete.PIDR\_REVAND**

TRCPIDR REVAND value.

Type: uint8\_t

Default value: 0

**subcluster0.ete.PIDR\_REVISION**

TRCPIDR REVISION value.



Type: `uint8_t`

Default value: 0

#### **`subcluster0.ete.Q_CADENCE`**

Number of instruction blocks traced between two Q elements.

Type: `uint16_t`

Default value: 0x1

#### **`subcluster0.ete.RES0_STATEFUL`**

Whether **RES0** bits are stateful or **RAZ/WI**.

Type: `bool`

Default value: `false`

#### **`subcluster0.ete.RETSTACK`**

Return stack depth.

Type: `uint8_t`

Default value: 3

#### **`subcluster0.ete.REVISION`**

TRCIDR1 revision value.

Type: `uint8_t`

Default value: 0

#### **`subcluster0.ete.SIM_OVERFLOW_GRANULARITY`**

Number of instruction blocks in each granule, for simulated overflow.

Type: `uint32_t`

Default value: 0x64

#### **`subcluster0.ete.SIM_OVERFLOW_PERCENTAGE`**

Percentage of instruction blocks lost in each granule, for simulated overflow.

Type: `uint8_t`

Default value: 0

#### **`subcluster0.ete.SOURCE_ADDRESS`**

Allow generation of source address elements.

Type: `bool`

Default value: `false`

### **`subcluster0.ete.TRACE_OUTPUT`**

File to which to write trace byte stream.

Type: `string`

Default value: `N/A`

### **`subcluster0.ete.TRCSRSTA_FORCED_EXCEP`**

TRCSRSTA value for a forcibly traced exception.

Type: `bool`

Default value: `false`

### **`subcluster0.force_mte_tag_access_razwi_and_ignore_tag_checks`**

Force MTE tag accesses to **RAZ/WI** and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

Type: `bool`

Default value: `false`

### **`subcluster0.force_zero_PSTATE_PAN`**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

### **`subcluster0.force_zero_mpam_partid_and_pmg`**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAMO\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

Type: `uint8_t`

Default value: `0`

### **`subcluster0.has_coherent_icache`**

Whether icache invalidation to the point of unification is required for instruction to data coherence.  
true - Invalidate operations not required.

Type: `bool`

Default value: `true`

#### **`subcluster0.has_delayed_dbgreg`**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

#### **`subcluster0.has_delayed_sysreg`**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

#### **`subcluster0.has_dot_product`**

Possible values of this parameter are:- 1, feature not implemented.- 2, feature is implemented.

Type: `uint8_t`

Default value: 2

#### **`subcluster0.has_ete`**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (`-plugin` or `-P`).

Type: `bool`

Default value: `false`

#### **`subcluster0.icache-hit_latency`**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.icache-maintenance_latency`**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

**subcluster0.icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.icache-size**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: `0x10000`

**subcluster0.invalidate\_code\_cache\_on\_icache\_cmo**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: `uint8_t`

Default value: 0

### **subcluster0.max\_32bit\_el**

Maximum exception level supporting AArch32 modes. -1: No Support for A32 at any EL, 0: EL0 supports A32. This parameter is ignored in Rev0 i.e. when `revision_number` = 0.

Type: `int8_t`

Default value: 0

### **subcluster0.memory\_tagging\_support\_level**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

Type: `uint8_t`

Default value: 3

### **subcluster0.ptw\_latency**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

### **subcluster0.ras\_pfg\_clock\_mhz**

RAS Pseudo-Fault generation clock rate in MHz.

Type: `uint8_t`

Default value: 12

### **subcluster0.revision\_number**

Revision number of TRM implemented by the model. Changing the value can change the model behaviour. Visible as the Variant field in MIDR/MIDR\_EL1. Corresponds to the revision number X in rXpY.

Type: `uint8_t`

Default value: 1

### **subcluster0.tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

#### **subcluster0.tlbi\_stall\_enabled**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: bool

Default value: false

#### **subcluster0.treat\_PAC\_as\_NOP**

Non-architecture parameter. Treat Pointer Authentication as **NOP**. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are **NOP**. Branch, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

Type: bool

Default value: false

#### **subcluster0.walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.NUM\_CORES**

Number of cores in the subcluster. Total number of cores in cluster may not exceed 12.

Type: uint8\_t

Default value: 1

#### **subcluster1.cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: uint32\_t

Default value: 0x1

#### **subcluster1.cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: uint32\_t

Default value: 0x1

**subcluster1.cpu0.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu0.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu0.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu0.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu0.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu0.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu0.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu0.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu0.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu0.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster1.cpu0.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu0.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t



Default value: 0x0

**subcluster1.cpu0.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu0.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu0.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu0.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu0.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu0.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu0.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu0.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster1.cpu0.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster1.cpu0.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu0.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu0.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu0.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu0.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu0.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu0.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu0.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu0.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu1.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu1.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu1.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu1.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

### **`subcluster1.cpu1.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu1.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu1.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu1.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu1.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu1.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu1.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu1.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**`subcluster1.cpu1.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**`subcluster1.cpu1.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**`subcluster1.cpu1.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**`subcluster1.cpu1.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**`subcluster1.cpu1.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**`subcluster1.cpu1.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**`subcluster1.cpu1.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster1.cpu1.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster1.cpu1.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu1.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu2.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster1.cpu2.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu2.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`



**subcluster1.cpu2.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter trace\_special\_hlt\_imm16.

Type: `bool`

Default value: `false`

**subcluster1.cpu2.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: 0x0

#### **subcluster1.cpu2.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

#### **subcluster1.cpu2.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu2.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu2.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu2.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu2.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu2.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu2.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu2.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu2.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu2.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster1.cpu2.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster1.cpu2.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster1.cpu2.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster1.cpu2.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu2.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu2.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu2.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu2.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu2.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu3.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster1.cpu3.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu3.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu3.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu3.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

**subcluster1.cpu3.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster1.cpu3.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu3.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu3.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu3.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu3.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu3.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster1.cpu3.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster1.cpu3.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster1.cpu3.semihosting-cwd**

Base directory for semihosting file access.

Type: string

Default value: N/A

**subcluster1.cpu3.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool



Default value: `true`

### **`subcluster1.cpu3.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu3.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster1.cpu3.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster1.cpu3.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster1.cpu3.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu3.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu4.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu4.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu4.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu4.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu4.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu4.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu4.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu4.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu4.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu4.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster1.cpu4.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu4.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu4.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu4.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu4.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu4.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu4.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu4.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu4.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu4.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster1.cpu4.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster1.cpu4.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu4.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu4.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu4.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu4.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu4.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu4.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu4.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu4.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu5.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu5.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu5.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu5.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu5.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu5.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu5.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu5.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu5.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu5.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

### **`subcluster1.cpu5.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu5.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu5.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`



**subcluster1.cpu5.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu5.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu5.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu5.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu5.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu5.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu5.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

### **`subcluster1.cpu5.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

### **`subcluster1.cpu5.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **`subcluster1.cpu5.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **`subcluster1.cpu5.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster1.cpu5.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu5.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster1.cpu5.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster1.cpu5.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster1.cpu5.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu5.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu6.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster1.cpu6.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu6.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu6.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter trace\_special\_hlt\_imm16.

Type: `bool`

Default value: `false`

**subcluster1.cpu6.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu6.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu6.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu6.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu6.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: 0x0

#### **subcluster1.cpu6.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

#### **subcluster1.cpu6.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu6.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu6.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu6.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu6.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu6.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu6.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu6.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu6.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu6.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster1.cpu6.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster1.cpu6.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster1.cpu6.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster1.cpu6.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu6.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu6.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu6.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu6.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu6.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster1.cpu6.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu7.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu7.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu7.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu7.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu7.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`



**subcluster1.cpu7.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu7.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu7.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu7.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu7.l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

**subcluster1.cpu7.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster1.cpu7.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu7.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu7.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu7.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu7.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu7.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu7.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu7.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu7.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster1.cpu7.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster1.cpu7.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster1.cpu7.semihosting-cwd**

Base directory for semihosting file access.

Type: string

Default value: N/A

**subcluster1.cpu7.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: `true`

#### **`subcluster1.cpu7.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu7.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

#### **`subcluster1.cpu7.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

#### **`subcluster1.cpu7.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

#### **`subcluster1.cpu7.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster1.cpu7.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu8.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu8.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu8.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu8.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu8.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu8.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu8.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu8.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu8.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu8.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster1.cpu8.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu8.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu8.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu8.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu8.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu8.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu8.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu8.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu8.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu8.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster1.cpu8.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster1.cpu8.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu8.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu8.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu8.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`



**subcluster1.cpu8.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu8.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu8.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu8.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu8.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu9.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu9.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu9.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu9.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu9.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu9.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu9.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu9.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu9.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu9.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

### **`subcluster1.cpu9.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu9.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu9.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu9.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu9.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu9.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu9.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu9.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu9.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

### **`subcluster1.cpu9.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

### **`subcluster1.cpu9.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **`subcluster1.cpu9.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **`subcluster1.cpu9.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster1.cpu9.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu9.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster1.cpu9.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster1.cpu9.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster1.cpu9.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu9.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster1.dcache-hit_latency`**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.dcache-maintenance_latency`**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.dcache-miss_latency`**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **subcluster1.dcache-prefetch\_enabled**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

Type: bool

Default value: false

### **subcluster1.dcache-read\_access\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster1.dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster1.dcache-size**

L1 D-Cache size in bytes.

Type: uint32\_t

Default value: 0x8000

### **subcluster1.dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster1.dcache-write\_access\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be

used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.dcache-write_latency`**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.ete.CLAIMTAGS`**

Number of claim tags.

Type: `uint8_t`

Default value: 32

#### **`subcluster1.ete.MAX_INST_PER_Q`**

Maximum limit for the number of instructions implied by a Q element.

Type: `uint16_t`

Default value: `0x1`

#### **`subcluster1.ete.NumberOfRSPairs`**

Number of resource selector pairs.

Type: `uint8_t`

Default value: 8

#### **`subcluster1.ete.PIDR_CM0D`**

TRCPIDR CM0D value.

Type: `uint8_t`

Default value: 0

#### **`subcluster1.ete.PIDR_REVAND`**

TRCPIDR REVAND value.

Type: `uint8_t`



Default value: 1

**subcluster1.ete.PIDR\_REVISION**

TRCPIDR REVISION value.

Type: `uint8_t`

Default value: 0

**subcluster1.ete.Q\_CADENCE**

Number of instruction blocks traced between two Q elements.

Type: `uint16_t`

Default value: 0x1

**subcluster1.ete.RES0\_STATEFUL**

Whether **RES0** bits are stateful or **RAZ/WI**.

Type: `bool`

Default value: `false`

**subcluster1.ete.RETSTACK**

Return stack depth.

Type: `uint8_t`

Default value: 3

**subcluster1.ete.REVISION**

TRCIDR1 revision value.

Type: `uint8_t`

Default value: 2

**subcluster1.ete.SIM\_OVERFLOW\_GRANULARITY**

Number of instruction blocks in each granule, for simulated overflow.

Type: `uint32_t`

Default value: 0x64

**subcluster1.ete.SIM\_OVERFLOW\_PERCENTAGE**

Percentage of instruction blocks lost in each granule, for simulated overflow.

Type: `uint8_t`

Default value: 0

**subcluster1.ete.SOURCE\_ADDRESS**

Allow generation of source address elements.

Type: bool

Default value: false

**subcluster1.ete.TRACE\_OUTPUT**

File to which to write trace byte stream.

Type: string

Default value: N/A

**subcluster1.ete.TRCRSRTA\_FORCED\_EXCEP**

TRCRSR.TA value for a forcibly traced exception.

Type: bool

Default value: false

**subcluster1.ext\_abort\_device\_read\_is\_sync**

Synchronous reporting of device-nGnRE read external aborts.

Type: bool

Default value: false

**subcluster1.ext\_abort\_device\_write\_is\_sync**

Synchronous reporting of device-nGnRE write external aborts.

Type: bool

Default value: false

**subcluster1.ext\_abort\_so\_read\_is\_sync**

Synchronous reporting of device-nGnRnE read external aborts.

Type: bool

Default value: false

**subcluster1.ext\_abort\_so\_write\_is\_sync**

Synchronous reporting of device-nGnRnE write external aborts.

Type: bool

Default value: `false`

#### **subcluster1.force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks**

Force MTE tag accesses to **RAZ/WI** and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

Type: `bool`

Default value: `false`

#### **subcluster1.force\_zero\_PSTATE\_PAN**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

#### **subcluster1.force\_zero\_mpam\_partid\_and\_pmg**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAMO\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

Type: `uint8_t`

Default value: `0`

#### **subcluster1.has\_coherent\_icache**

Whether icache invalidation to the point of unification is required for instruction to data coherence.  
true - Invalidate operations not required.

Type: `bool`

Default value: `true`

#### **subcluster1.has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

#### **subcluster1.has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

### **`subcluster1.has_ete`**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (`-plugin` or `-P`).

Type: `bool`

Default value: `false`

### **`subcluster1.icache-hit_latency`**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.icache-maintenance_latency`**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.icache-miss_latency`**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.icache-prefetch_enabled`**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

### **`subcluster1.icache-read_access_latency`**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.icache-read_latency`**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.icache-size`**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: `0x8000`

### **`subcluster1.invalidate_code_cache_on_icache_cmo`**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: `uint8_t`

Default value: `0`

### **`subcluster1.memory_tagging_support_level`**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).

Type: `uint8_t`

Default value: `2`

### **`subcluster1.ptw_latency`**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.tlb_latency`**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

**subcluster1.tlbi\_stall\_enabled**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: bool

Default value: false

**subcluster1.treat-dcache-cmos-to-pou-as-nop**

Whether dcache invalidation to the point of unification is required for instruction to data coherence. 0 - Invalidate ops required, 1 - Invalidate ops not required and cannot generate faults, 2 - Invalidate ops not required but can generate faults.

Type: uint8\_t

Default value: 0

**subcluster1.walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

**subcluster2.CPUCFR**

Value of CPU Configuration Register.

Type: uint64\_t

Default value: 0x0

**subcluster2.NUM\_CORES**

Number of cores per cluster.

Type: uint8\_t

Default value: 1

**subcluster2.cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: uint32\_t

Default value: 0x1

**subcluster2.cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**subcluster2.cpu0.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster2.cpu0.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster2.cpu0.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster2.cpu0.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu0.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster2.cpu0.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

### **`subcluster2.cpu0.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu0.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu0.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu0.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu0.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu0.l2cache-size`**

L2 Cache size in bytes.



Type: `uint32_t`

Default value: `0x100000`

#### **`subcluster2.cpu0.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster2.cpu0.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster2.cpu0.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster2.cpu0.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster2.cpu0.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

#### **`subcluster2.cpu0.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: 0

**`subcluster2.cpu0.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**`subcluster2.cpu0.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**`subcluster2.cpu0.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: 0x123456

**`subcluster2.cpu0.semihosting-T32_HLT`**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**`subcluster2.cpu0.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**`subcluster2.cpu0.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**`subcluster2.cpu0.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

### **`subcluster2.cpu0.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster2.cpu0.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu0.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster2.cpu0.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster2.cpu0.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster2.cpu0.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster2.cpu0.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster2.cpu0.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster2.cpu1.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster2.cpu1.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster2.cpu1.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster2.cpu1.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu1.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu1.force-fpsid`**

Override the FPSID value.

Type: `bool`

Default value: `true`

### **`subcluster2.cpu1.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu1.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu1.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu1.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu1.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to

the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster2.cpu1.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x100000`

#### **`subcluster2.cpu1.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster2.cpu1.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster2.cpu1.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster2.cpu1.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu1.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster2.cpu1.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster2.cpu1.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu1.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu1.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster2.cpu1.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster2.cpu1.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster2.cpu1.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster2.cpu1.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster2.cpu1.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster2.cpu1.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu1.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu1.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster2.cpu1.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`



Default value: 0xf000000

#### **subcluster2.cpu1.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

#### **subcluster2.cpu1.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

#### **subcluster2.cpu1.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: true

#### **subcluster2.cpu2.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: false

#### **subcluster2.cpu2.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: bool

Default value: false

#### **subcluster2.cpu2.CRYPTODISABLE**

Disable cryptographic features.

Type: bool

Default value: false

**subcluster2.cpu2.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu2.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster2.cpu2.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster2.cpu2.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu2.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu2.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu2.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu2.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu2.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x100000

**subcluster2.cpu2.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu2.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu2.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu2.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu2.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

### **`subcluster2.cpu2.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

### **`subcluster2.cpu2.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster2.cpu2.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster2.cpu2.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster2.cpu2.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster2.cpu2.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster2.cpu2.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster2.cpu2.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster2.cpu2.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster2.cpu2.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: 0x0

**subcluster2.cpu2.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: 0xf000000

**subcluster2.cpu2.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster2.cpu2.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu2.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster2.cpu2.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster2.cpu2.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster2.cpu3.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster2.cpu3.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster2.cpu3.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster2.cpu3.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu3.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster2.cpu3.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster2.cpu3.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu3.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu3.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu3.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu3.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu3.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x100000`

### **`subcluster2.cpu3.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`



**subcluster2.cpu3.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu3.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu3.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu3.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster2.cpu3.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster2.cpu3.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu3.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster2.cpu3.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster2.cpu3.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster2.cpu3.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster2.cpu3.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster2.cpu3.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster2.cpu3.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster2.cpu3.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu3.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster2.cpu3.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster2.cpu3.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster2.cpu3.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu3.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster2.cpu3.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster2.cpu4.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster2.cpu4.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

#### **`subcluster2.cpu4.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster2.cpu4.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster2.cpu4.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

#### **`subcluster2.cpu4.force-fpsid`**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster2.cpu4.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu4.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu4.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu4.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu4.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu4.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x100000

#### **subcluster2.cpu4.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster2.cpu4.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster2.cpu4.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster2.cpu4.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster2.cpu4.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

#### **subcluster2.cpu4.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster2.cpu4.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster2.cpu4.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster2.cpu4.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster2.cpu4.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster2.cpu4.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster2.cpu4.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster2.cpu4.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster2.cpu4.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: true

**subcluster2.cpu4.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu4.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster2.cpu4.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster2.cpu4.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster2.cpu4.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000



**subcluster2.cpu4.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster2.cpu4.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster2.cpu5.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster2.cpu5.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster2.cpu5.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster2.cpu5.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu5.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu5.force-fpsid`**

Override the FPSID value.

Type: `bool`

Default value: `true`

### **`subcluster2.cpu5.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu5.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu5.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu5.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu5.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to

the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster2.cpu5.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x100000`

#### **`subcluster2.cpu5.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster2.cpu5.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster2.cpu5.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster2.cpu5.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu5.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster2.cpu5.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster2.cpu5.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu5.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu5.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster2.cpu5.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster2.cpu5.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster2.cpu5.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster2.cpu5.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster2.cpu5.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster2.cpu5.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu5.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu5.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster2.cpu5.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: 0xf000000

**subcluster2.cpu5.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu5.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster2.cpu5.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: true

**subcluster2.cpu6.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: false

**subcluster2.cpu6.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: bool

Default value: false

**subcluster2.cpu6.CRYPTODISABLE**

Disable cryptographic features.

Type: bool

Default value: false

**subcluster2.cpu6.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu6.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster2.cpu6.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster2.cpu6.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu6.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu6.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu6.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu6.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu6.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x100000

**subcluster2.cpu6.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu6.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu6.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.



Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu6.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu6.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

### **`subcluster2.cpu6.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

### **`subcluster2.cpu6.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster2.cpu6.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster2.cpu6.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster2.cpu6.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster2.cpu6.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster2.cpu6.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster2.cpu6.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster2.cpu6.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster2.cpu6.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: 0x0

**subcluster2.cpu6.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: 0xf000000

**subcluster2.cpu6.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster2.cpu6.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu6.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster2.cpu6.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster2.cpu6.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster2.cpu7.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster2.cpu7.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster2.cpu7.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster2.cpu7.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu7.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster2.cpu7.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster2.cpu7.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu7.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu7.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu7.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu7.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu7.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x100000`

### **`subcluster2.cpu7.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu7.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu7.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu7.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu7.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster2.cpu7.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster2.cpu7.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu7.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster2.cpu7.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster2.cpu7.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster2.cpu7.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster2.cpu7.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster2.cpu7.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster2.cpu7.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster2.cpu7.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu7.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu7.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster2.cpu7.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu7.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster2.cpu7.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster2.cpu7.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).



Type: `bool`

Default value: `true`

#### **`subcluster2.cpu8.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster2.cpu8.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

#### **`subcluster2.cpu8.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster2.cpu8.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster2.cpu8.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

#### **`subcluster2.cpu8.force-fpsid`**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster2.cpu8.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu8.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu8.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu8.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu8.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu8.l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: 0x100000

**subcluster2.cpu8.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu8.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu8.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu8.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu8.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster2.cpu8.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster2.cpu8.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster2.cpu8.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster2.cpu8.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: 0x123456

**subcluster2.cpu8.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster2.cpu8.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster2.cpu8.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster2.cpu8.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster2.cpu8.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: true

**subcluster2.cpu8.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu8.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster2.cpu8.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster2.cpu8.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster2.cpu8.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu8.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster2.cpu8.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster2.cpu9.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster2.cpu9.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster2.cpu9.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster2.cpu9.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu9.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu9.force-fpsid`**

Override the FPSID value.

Type: `bool`

Default value: `true`

### **`subcluster2.cpu9.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu9.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu9.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu9.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu9.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to

the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster2.cpu9.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x100000`

#### **`subcluster2.cpu9.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster2.cpu9.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster2.cpu9.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster2.cpu9.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`



**subcluster2.cpu9.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster2.cpu9.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster2.cpu9.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu9.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu9.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster2.cpu9.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster2.cpu9.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster2.cpu9.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster2.cpu9.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster2.cpu9.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster2.cpu9.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu9.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu9.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster2.cpu9.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: 0xf000000

#### **subcluster2.cpu9.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

#### **subcluster2.cpu9.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

#### **subcluster2.cpu9.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: true

#### **subcluster2.dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster2.dcache-maintenance\_latency**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster2.dcache-miss\_latency**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster2.dcache-prefetch\_enabled**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

Type: bool

Default value: false

### **subcluster2.dcache-read\_access\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster2.dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster2.dcache-size**

L1 D-Cache size in bytes.

Type: uint32\_t

Default value: 0x8000

### **subcluster2.dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster2.dcache-write\_access\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be

used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **subcluster2.dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **subcluster2.error\_record\_feature\_register**

RAS feature register values. An array of JSON objects. The JSON schema for the array is: `[{"ED":0x0, "IMPDEF_3_2":0x0, "UI":0x0, "FI":0x0, "UE":0x0, "CFI":0x0, "CEC":0x0, "RP":0x0, "DUI":0x0, "CEO":0x0, "CI":0x0, "TS":0x0, "INJ":0x0, "FRX":0x0, "UC":0x0, "UEU":0x0, "UER":0x0, "UEO":0x0, "DE":0x0, "CE":0x0, "Visibility":"Core"},other_feature_register_values]`. Where ED,UI,FI,CE and UE have valid values between `0x0 - 0x3`. CFI and DUI have valid values `0x0, 0x2` and `0x3`. CEC has valid values `0x0,0x2` or `0x4`. RP,CEO,INJ,FRX,UC,UEU,UER,UEO,DE has valid values `0x0` or `0x1`. CI and TS has valid values of `0x0, 0x1` and `0x2`. Visibility has valid values "Core" or "Cluster".

Type: `string`

Default value: `"[{\"ED\":0x2,\"IMPDEF_3_2\":0x0,\"UI\":0x2,\"FI\":0x2,\"UE\":0x1,\"CFI\":0x2,\"CEC\":0x2,\"RP\":0x1,\"DUI\":0x0,\"CEO\":0x0,\"INJ\":0x1,\"CI\":0x0,\"TS\":0x0,\"Visibility\":\"Cluster\"},{\"ED\":0x2,\"IMPDEF_3_2\":0x0,\"UI\":0x2,\"FI\":0x2,\"UE\":0x1,\"CFI\":0x2,\"CEC\":0x2,\"RP\":0x1,\"DUI\":0x0,\"CEO\":0x0,\"INJ\":0x1,\"CI\":0x0,\"TS\":0x0}]]"`

### **subcluster2.ete.CLAIMTAGS**

Number of claim tags.

Type: `uint8_t`

Default value: `32`

### **subcluster2.ete.MAX\_INST\_PER\_Q**

Maximum limit for the number of instructions implied by a Q element.

Type: `uint16_t`

Default value: `0x1`

**subcluster2.ete.NumberOfRSPairs**

Number of resource selector pairs.

Type: `uint8_t`

Default value: 8

**subcluster2.ete.PIDR\_CM0D**

TRCPIDR CM0D value.

Type: `uint8_t`

Default value: 0

**subcluster2.ete.PIDR\_REVAND**

TRCPIDR REVAND value.

Type: `uint8_t`

Default value: 0

**subcluster2.ete.PIDR\_REVISION**

TRCPIDR REVISION value.

Type: `uint8_t`

Default value: 0

**subcluster2.ete.Q\_CADENCE**

Number of instruction blocks traced between two Q elements.

Type: `uint16_t`

Default value: 0x1

**subcluster2.ete.RES0\_STATEFUL**

Whether **RES0** bits are stateful or **RAZ/WI**.

Type: `bool`

Default value: `false`

**subcluster2.ete.RETSTACK**

Return stack depth.

Type: `uint8_t`

Default value: 3

**subcluster2.ete.REVISION**

TRCIDR1 revision value.

Type: `uint8_t`

Default value: 0

**subcluster2.ete.SIM\_OVERFLOW\_GRANULARITY**

Number of instruction blocks in each granule, for simulated overflow.

Type: `uint32_t`

Default value: 0x64

**subcluster2.ete.SIM\_OVERFLOW\_PERCENTAGE**

Percentage of instruction blocks lost in each granule, for simulated overflow.

Type: `uint8_t`

Default value: 0

**subcluster2.ete.SOURCE\_ADDRESS**

Allow generation of source address elements.

Type: `bool`

Default value: `false`

**subcluster2.ete.TRACE\_OUTPUT**

File to which to write trace byte stream.

Type: `string`

Default value: N/A

**subcluster2.ete.TRCRSRTA\_FORCED\_EXCEP**

TRCRSR.TA value for a forcibly traced exception.

Type: `bool`

Default value: `false`

**subcluster2.ext\_abort\_normal\_noncacheable\_read\_is\_sync**

Synchronous reporting of normal noncacheable-read external aborts.

Type: `bool`

Default value: `true`

**subcluster2.ext\_abort\_so\_write\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO , 4 = UER, 5 = CE.

Type: `uint8_t`

Default value: 2

**subcluster2.force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks**

Force MTE tag accesses to **RAZ/WI** and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

Type: `bool`

Default value: `false`

**subcluster2.force\_zero\_PSTATE\_PAN**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

**subcluster2.force\_zero\_mpam\_partid\_and\_pmg**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

Type: `uint8_t`

Default value: 0

**subcluster2.has\_coherent\_icache**

Whether icache invalidation to the point of unification is required for instruction to data coherence. true - Invalidate operations not required.

Type: `bool`

Default value: `true`

**subcluster2.has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`



Default value: `true`

### **`subcluster2.has_delayed_sysreg`**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

### **`subcluster2.has_enhanced_pan`**

Implements Armv8.7 Enhanced PAN feature (FEAT\_PAN3) values of this parameter are:- 1, feature is implemented if ARMv8.7 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 2

### **`subcluster2.has_ete`**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (-plugin or -P).

Type: `bool`

Default value: `false`

### **`subcluster2.has_large_va`**

Implement support for the extended 52-bit virtual addresses from ARMv8.2 (FEAT\_LVA). values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.2 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **`subcluster2.icache-hit_latency`**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **`subcluster2.icache-maintenance_latency`**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **subcluster2.icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster2.icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

Type: bool

Default value: false

### **subcluster2.icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster2.icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster2.icache-size**

L1 I-Cache size in bytes.

Type: uint32\_t

Default value: 0x8000

### **subcluster2.instruction\_tlb\_size**

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

Type: uint32\_t

Default value: 0x0

### **subcluster2.invalidate\_code\_cache\_on\_icache\_cmo**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: uint8\_t

Default value: 0

### **subcluster2.memory\_tagging\_support\_level**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

Type: uint8\_t

Default value: 3

### **subcluster2.pmu-num\_counters**

Number of PMU counters implemented.

Type: uint8\_t

Default value: 6

### **subcluster2.pseudo\_fault\_generation\_feature\_register**

ARMv8.4 Standard Pseudo-fault generation feature register values. JSON schema for the parameter value is: [{"OF":false, "UC":false, "UEU":false, "UER":false, "UEO":false, "DE":false, "CE":0x0, "CI":false, "ER":false, "PN":false, "AV":false, "MV":false, "SYN":false, "R":false, "NA":false}, other\_psuedo-fault\_generating\_features\_register\_values]. Where OF, UC, UEU, UER, UEO, DE, CI, ER, PN, AV, MV, SYN, and R have valid false(NOT\_SUPPORTED) and true(FEATURE\_CONTROLLABLE), where CE can have 0(NOT\_SUPPORTED), 1(NONSPECIFIC\_CE\_SUPPORTED) and 3(TRANSIENT\_OR\_PERSISTENT\_CE\_SUPPORTED) and NA can have false(component fakes detection on next access) or true(component fakes detection spontaneously). Effective only when ERXFR's INJ field allows it or has\_ras\_fault\_injection is true.

Type: string

Default value: "[{"OF":true, "UC":true, "UEU":false, "UER":false, "UEO":false, "DE":0x1, "CE":0x1, "CI":true, "ER":false, "PN":true, "AV":false, "MV":true, "SYN":true, "R":true}, {"OF":false, "UC":true, "UEU":false, "UER":false, "UEO":false, "DE":0x1, "CE":0x1, "CI":false, "ER":false, "PN":false, "AV":false, "MV":false, "SYN":false, "R":true}]"

**subcluster2.ptw\_latency**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.stage12\_tlb\_size**

Number of stage1+2 tlb entries. Valid values are 0 or  $\geq 4$ .

Type: `uint32_t`

Default value: `0x80`

**subcluster2.tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.tlbi\_stall\_enabled**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: `bool`

Default value: `false`

**subcluster2.trace\_physical\_registers\_when\_host\_virtualisation\_enabled**

Trace sysreg accesses to physical register accessed (0=disabled, 1=Trace only ELR/SPSR\_EL1 as ELR/SPSR\_EL2, 2=Trace all redirected registers as physical registers affecting all features inc. the host virtualised registers).

Type: `uint8_t`

Default value: `1`

**subcluster2.treat\_PAC\_as\_NOP**

Non-architecture parameter. Treat Pointer Authentication as **NOP**. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are **NOP**. Branch, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

Type: `bool`

Default value: `false`

**subcluster2.walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

## 3.48 ARMCortexA510CT\_CortexA715CT\_CortexX3CT

Defined in LISA/ARMCortexA510CT\_CortexA715CT\_CortexX3CT.lisa.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
CortexA510 r0p0	Full support
CortexA510 r1p3	Full support
CortexA715 r1p2	Full support
CortexX3 r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following parameters were added:

- num\_acp
- subcluster0.has\_delayed\_dbgreg
- subcluster0.has\_delayed\_sysreg
- subcluster1.has\_delayed\_dbgreg
- subcluster1.has\_delayed\_sysreg
- subcluster2.has\_delayed\_dbgreg
- subcluster2.has\_delayed\_sysreg

### About ARMCortexA510CT\_CortexA715CT\_CortexX3CT

The number of cores in each subcluster is configurable using the following parameters:

**subcluster0.NUM\_CORES**

Possible values are 1-10 (ARMCortexA510CT).

**subcluster1.NUM\_CORES**

Possible values are 1-10 (ARMCortexA715CT).

**subcluster2.NUM\_CORES**

Possible values are 1-10 (ARMCortexX3CT).

The total number of cores in the cluster cannot exceed 12.

Port arrays are expanded to the width needed for the maximum number of cores in each subcluster. Use the following port array indexes:

- `<port_name>[0-9]` for cores in `subcluster0`.
- `<port_name>[10-19]` for cores in `subcluster1`.
- `<port_name>[20-29]` for cores in `subcluster2`.



Note

All instances in the Master cross trigger matrix port array, `cti[30]` must be connected, regardless of the `NUM_CORES` value used.

Core-specific parameters have the following prefixes:

- `subcluster0.cpu0` to `subcluster0.cpu9` identify cores in `subcluster0`.
- `subcluster1.cpu0` to `subcluster1.cpu9` identify cores in `subcluster1`.
- `subcluster2.cpu0` to `subcluster2.cpu9` identify cores in `subcluster2`.

For information about the cores in this model, see:

- [ARMCortexA510CT](#).
- [ARMCortexA715CT](#).
- [ARMCortexX3CT](#).

## Iris and MTI instances for ARMCortexA510CT\_CortexA715CT\_CortexX3CT

This model has the following Iris instances:

Name	Instance type
ARMCortexA510CT_CortexA715CT_CortexX3CT	Cluster_ARM_CortexA510_CortexA715_CortexX3_Heterogeneous
ARMCortexA510CT_CortexA715CT_CortexX3CT.AMU	PVBusLogger
ARMCortexA510CT_CortexA715CT_CortexX3CT.AMU.mapper	PVBusMapper
ARMCortexA510CT_CortexA715CT_CortexX3CT.DAP	PVBusLogger
ARMCortexA510CT_CortexA715CT_CortexX3CT.DAP.mapper	PVBusMapper
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU	DSU-110
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU.PPU_cluster	PPUv1
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU.PPU_coreU (where $U = 0-2$ )	PPUv1
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU.PPU_coreU.busslave (where $U = 0-2$ )	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU.l3_flusher	AsyncCacheFlushUnit

Name	Instance type
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU.shared_cache	PVCache
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU.shared_cache.upstream[U] (where $U = 0-7$ )	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.MMAP	PVBusLogger
ARMCortexA510CT_CortexA715CT_CortexX3CT.MMAP.mapper	PVBusMapper
ARMCortexA510CT_CortexA715CT_CortexX3CT.RAS	PVBusLogger
ARMCortexA510CT_CortexA715CT_CortexX3CT.RAS.mapper	PVBusMapper
ARMCortexA510CT_CortexA715CT_CortexX3CT.cpuU.debug_rom (where $U = 0-2$ )	debug_rom
ARMCortexA510CT_CortexA715CT_CortexX3CT.ext_bus	PVBusLogger
ARMCortexA510CT_CortexA715CT_CortexX3CT.ext_bus.mapper	PVBusMapper
ARMCortexA510CT_CortexA715CT_CortexX3CT.gic_cpuif_decoder_cluster	GLCv3CPUInterfaceDecoder
ARMCortexA510CT_CortexA715CT_CortexX3CT.global_debug_rom	debug_rom
ARMCortexA510CT_CortexA715CT_CortexX3CT.secondary_debug_rom	debug_rom
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster0	Subcluster_ARM_Cortex-A510
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster0.cpu0	ARM_Cortex-A510
ARMCortexA510CT_CortexA715CT_CortexX3CT.subclusterU.cpu0.UTLB (where $U = 0-2$ )	TLB
ARMCortexA510CT_CortexA715CT_CortexX3CT.subclusterU.cpuV.dtlb (where $U = 0-2$ ; $V = 0-2$ )	TLB
ARMCortexA510CT_CortexA715CT_CortexX3CT.subclusterU.cpu0.l1dcache (where $U = 0-2$ )	PVCache
ARMCortexA510CT_CortexA715CT_CortexX3CT.subclusterU.cpu0.l1dcache.downstream[0].pvbusmaster (where $U = 0-2$ )	PVBusMaster
ARMCortexA510CT_CortexA715CT_CortexX3CT.subclusterU.cpu0.l1dcache.upstream[0] (where $U = 0-2$ )	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.subclusterU.cpu0.l1licache (where $U = 0-2$ )	PVCache

Name	Instance type
ARMCortexA510CT_CortexA715CT_CortexX3CT.subclusterU.cpu0.l1icache.downstream[0].pvbusmaster (where $U = 0-2$ )	PVBusMaster
ARMCortexA510CT_CortexA715CT_CortexX3CT.subclusterU.cpu0.l1icache.upstream[0] (where $U = 0-2$ )	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.subclusterU.cpu0.l2cache (where $U = 0-2$ )	PVCache
ARMCortexA510CT_CortexA715CT_CortexX3CT.subclusterU.cpu0.l2cache.downstream[0].pvbusmaster (where $U = 0-2$ )	PVBusMaster
ARMCortexA510CT_CortexA715CT_CortexX3CT.subclusterU.cpu0.l2cache.upstream[A] (where $U = 0-2$ ; $A = 0-1$ )	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.subclusterU.sve (where $U = 0-2$ )	ScalableVectorExtension
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster1	Subcluster_ARM_Cortex-A715
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster1.cpu0	ARM_Cortex-A715
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster2	Subcluster_ARM_Cortex-X3
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster2.cpu0	ARM_Cortex-X3

This model has the following MTI trace components:

Name	Component type
ARMCortexA510CT_CortexA715CT_CortexX3CT.AMU	PVBusLogger
ARMCortexA510CT_CortexA715CT_CortexX3CT.AMU.mapper	PVBusMapper
ARMCortexA510CT_CortexA715CT_CortexX3CT.DAP	PVBusLogger
ARMCortexA510CT_CortexA715CT_CortexX3CT.DAP.mapper	PVBusMapper
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU	DSU-110
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU.PPU_cluster	PPUv1
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU.PPU_coreU (where $U = 0-2$ )	PPUv1
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU.PPU_coreU.busslave (where $U = 0-2$ )	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU.shared_cache	PVCache
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU.shared_cache.upstream[U] (where $U = 0-7$ )	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.DSU.utility_slave[0]	PVBusSlave



Name	Component type
ARMCortexA510CT_CortexA715CT_CortexX3CT.MMAP	PVBusLogger
ARMCortexA510CT_CortexA715CT_CortexX3CT.MMAP.mapper	PVBusMapper
ARMCortexA510CT_CortexA715CT_CortexX3CT.RAS	PVBusLogger
ARMCortexA510CT_CortexA715CT_CortexX3CT.RAS.mapper	PVBusMapper
ARMCortexA510CT_CortexA715CT_CortexX3CT.ext_bus	PVBusLogger
ARMCortexA510CT_CortexA715CT_CortexX3CT.ext_bus.mapper	PVBusMapper
ARMCortexA510CT_CortexA715CT_CortexX3CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster0.cpu0	ARM_Cortex-A510
ARMCortexA510CT_CortexA715CT_CortexX3CT.subclusterU.cpu0.UTLB (where $U = 0-2$ )	TLB
ARMCortexA510CT_CortexA715CT_CortexX3CT.subclusterU.cpu0.l1dcache (where $U = 0-2$ )	PVCache
ARMCortexA510CT_CortexA715CT_CortexX3CT.subclusterU.cpu0.l1dcache.downstream[0].pvbusmaster (where $U = 0-2$ )	PVBusMaster
ARMCortexA510CT_CortexA715CT_CortexX3CT.subclusterU.cpu0.l1dcache.upstream[0] (where $U = 0-2$ )	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.subclusterU.cpu0.l1icache (where $U = 0-2$ )	PVCache
ARMCortexA510CT_CortexA715CT_CortexX3CT.subclusterU.cpu0.l1icache.downstream[0].pvbusmaster (where $U = 0-2$ )	PVBusMaster
ARMCortexA510CT_CortexA715CT_CortexX3CT.subclusterU.cpu0.l1icache.upstream[0] (where $U = 0-2$ )	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.subclusterU.cpu0.l2cache (where $U = 0-2$ )	PVCache
ARMCortexA510CT_CortexA715CT_CortexX3CT.subclusterU.cpu0.l2cache.downstream[0].pvbusmaster (where $U = 0-2$ )	PVBusMaster
ARMCortexA510CT_CortexA715CT_CortexX3CT.subclusterU.cpu0.l2cache.upstream[A] (where $U = 0-2$ ; $A = 0-1$ )	PVBusSlave
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster1.cpu0	ARM_Cortex-A715
ARMCortexA510CT_CortexA715CT_CortexX3CT.subcluster2.cpu0	ARM_Cortex-X3

### Ports for ARMCortexA510CT\_CortexA715CT\_CortexX3CT

Port	Direction	Protocol	Description
acp_s	slave	PVBus	AXI ACP slave port.
AEND0MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND2MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND3MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART2MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).

Port	Direction	Protocol	Description
ASTART3MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	slave	Signal	CHI defined pins.
broadcastcachemaint	slave	Signal	ACE defined pins.
broadcastouter	slave	Signal	ACE defined pins.
broadcastpersist	slave	Signal	CHI defined pins.
cfgend	slave	Signal	This signal is for EE bit initialisation.
cfgsdisable	slave	Signal	This signal disables write access to some secure Interrupt Controller registers.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcs_m_pchannel	master	PChannel	Cluster PCSM signal
cluster_powerdown_out	master	Signal	DEPRECATED - An output from PPU that informs thermal controller of the cluster power info
cluster_ppu_hw_stat	master	Value	Cluster PPU power state
clusterid	slave	Value	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set ManagerID64 of the core.
clusterpmuirq	master	Signal	DynamiQ pmu irq
CNTHPIRQ	master	Signal	Timer signals to SOC.
CNTHPSIRQ	master	Signal	Timer signals to SOC
CNTHVIRQ	master	Signal	Timer signals to SOC.
CNTHVSIQ	master	Signal	Timer signals to SOC
CNTPNSIRQ	master	Signal	Timer signals to SOC.
CNTPSIQ	master	Signal	Timer signals to SOC.
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.
CNTVIRQ	master	Signal	Timer signals to SOC.
commirq	master	Signal	Interrupt signal from debug communications channel.
core_clk_in	slave	ClockSignal	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
core_pcs_m_pchannel	master	PChannel	Core PCSM signals
core_powerdown_out	master	Signal	DEPRECATED - An output from PPU that informs thermal controller of the core power info
core_ppu_hw_stat	master	Value	Notify the power state change inside the PPU.

Port	Direction	Protocol	Description
coreerrirq	master	Signal	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq	master	Signal	Fault indicator for a detected 1 or 2 bit ECC error
coreinstrrun	master	Signal	Core Running State
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
ctidbgirq	master	Signal	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	slave	Signal	External debug interface.
dbgnopwrdown	master	Signal	No power-down request.
dbgppwrupreq	master	Signal	Debug power up request.
defaultMP	slave	Signal	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main master interface ports.
dev_debug_s	slave	PVBus	External debug interface.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.
gicreset	master	Signal	An output from PPU that can be used to reset GICCLK domain components
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	slave	PVBus	External debug interface.
pmbirq	master	Signal	Interrupt signal from the statistical profiling unit.
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	master	Signal	PPU cluster interrupt.
ppu_cluster_isolate	master	Signal	PPU Cluster Domain isolation control.
ppu_cluster_wakerequest	slave	Signal	PPU cluster wake request signal.
ppu_core_irq	master	Signal	PPU core interrupt.
ppu_core_isolate	master	Signal	PPU Core Domain isolation control.
ppu_core_wakerequest	slave	Signal	PPU Core wake request signals.
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	master	PVBus	The core will generate bus requests on this port.
pvbus_periph_m	master	PVBus	The core can generate peripheral bus request on this port.
rei	slave	Signal	Per core RAM Error Interrupt.
reset	slave	Signal	Reset signal to cluster.
rvbaraddr	slave	Value_64	Reset vector base address.
sci_m	master	SystemCoherencyInterface	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei	slave	Signal	Per core System Error physical pins.

Port	Direction	Protocol	Description
spiden	slave	Signal	External debug interface.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq	master	Signal	Interrupt signal from the trace buffer unit.
utility_bus_s	slave	PVBus	Utility bus slave
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfiq	slave	Signal	Virtualised FIQ.
virq	slave	Signal	Virtualised IRQ.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Per core virtual System Error physical pins.

## Parameters for ARM CortexA510CT\_CortexA715CT\_CortexX3CT

### AEND0\_DEFAULT

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

Type: uint64\_t

Default value: 0x0

### AEND1\_DEFAULT

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

Type: uint64\_t

Default value: 0x0

### AEND2\_DEFAULT

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

Type: uint64\_t

Default value: 0x0

### AEND3\_DEFAULT

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

Type: uint64\_t

Default value: 0x0

#### **ASTART0\_DEFAULT**

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

Type: `uint64_t`

Default value: 0x0

#### **ASTART1\_DEFAULT**

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

Type: `uint64_t`

Default value: 0x0

#### **ASTART2\_DEFAULT**

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

Type: `uint64_t`

Default value: 0x0

#### **ASTART3\_DEFAULT**

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

Type: `uint64_t`

Default value: 0x0

#### **BROADCASTATOMIC**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`

Default value: `true`

#### **BROADCASTCACHEMAINT**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

Type: `bool`

Default value: `false`

### **BROADCASTOUTER**

Enable broadcasting of Outer Shareable transactions. The `broadcastouter` signal will override this value if used.

Type: `bool`

Default value: `false`

### **BROADCASTPERSIST**

Enable broadcasting of cache clean to the point of persistence operations. The `broadcastpersist` signal will override this value if used.

Type: `bool`

Default value: `true`

### **CLUSTER\_ID**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the ManagerID64 of the core.

Type: `uint32_t`

Default value: `0x0`

### **CMO\_broadcast\_when\_cache\_state\_modelling\_disabled**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. This value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required.

Type: `uint8_t`

Default value: `1`

### **GICDISABLE**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`

Default value: `true`

### **bus\_type**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

Type: `uint8_t`

Default value: 0

### **core\_power\_on\_by\_default**

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPUs which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

Type: `bool`

Default value: `false`

### **dcache-state\_modelled**

Set whether D-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **default\_opmode**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

Type: `uint8_t`

Default value: 4

### **diagnostics**

Enable DynamIQ diagnostic messages.

Type: `bool`

Default value: `false`

### **enable\_simulation\_performance\_optimizations**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of `stage12_tlb_size` parameter to 1024).

Type: `bool`

Default value: `true`

### **has\_peripheral\_port**

If true, an additional AXI peripheral port is configured.

Type: `bool`

Default value: `false`

### **icache-state\_modelled**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **l3cache-has\_mpam**

L3 Cache has MPAM support.

Type: `bool`

Default value: `false`

### **l3cache-hit\_latency**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **l3cache-maintenance\_latency**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **l3cache-miss\_latency**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`



**l3cache-read\_access\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size l3cache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**l3cache-read\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. l3cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**l3cache-size**

L3 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**l3cache-snoop\_data\_transfer\_latency**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**l3cache-snoop\_issue\_latency**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**l3cache-ways**

L3 Cache number of ways (sets are implicit from size).

Type: uint16\_t

Default value: 16

### **l3cache-write\_access\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size l3cache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **l3cache-write\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. l3cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **mpmm\_accumulator\_multiplier**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator.value of n means the accumulator will use (n \* accumulator value) to calculate the mpmm threshold (MPMM).is provided as a fast model workaround to handle cases where execution of quantum in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

Type: uint8\_t

Default value: 1

### **num\_acp**

Number of ACP ports.

Type: uint8\_t

Default value: 0

### **num\_nodes**

Number of transport nodes. Zero implies direct-connect configuration.

Type: uint8\_t

Default value: 1

### **subcluster0.CPUCFR**

Value of CPU Configuration Register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.NUM_CORES`**

Number of cores in the subcluster. Total number of cores in cluster may not exceed 12.

Type: `uint8_t`

Default value: `1`

### **`subcluster0.cpi_div`**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

### **`subcluster0.cpi_mul`**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

### **`subcluster0.cpu0.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu0.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu0.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu0.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter trace\_special\_hlt\_imm16.

Type: bool

Default value: false

**subcluster0.cpu0.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu0.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu0.l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

**subcluster0.cpu0.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu0.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu0.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: `uint8_t`

Default value: `8`

**subcluster0.cpu0.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu0.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu0.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu0.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu0.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu0.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu0.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**`subcluster0.cpu0.semihosting-T32_HLT`**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**`subcluster0.cpu0.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**`subcluster0.cpu0.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**`subcluster0.cpu0.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**`subcluster0.cpu0.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**`subcluster0.cpu0.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**`subcluster0.cpu0.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf0000000`

### **`subcluster0.cpu0.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x100000000`

### **`subcluster0.cpu0.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf0000000`

### **`subcluster0.cpu0.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu0.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu0.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu1.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`



Default value: `false`

**subcluster0.cpu1.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster0.cpu1.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu1.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu1.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu1.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu1.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu1.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu1.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu1.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster0.cpu1.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

### **subcluster0.cpu1.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster0.cpu1.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

### **subcluster0.cpu1.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster0.cpu1.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

### **subcluster0.cpu1.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu1.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu1.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu1.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu1.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster0.cpu1.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster0.cpu1.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu1.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu1.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu1.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu1.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu1.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu1.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu1.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu1.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu1.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu2.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu2.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu2.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu2.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu2.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu2.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu2.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu2.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu2.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: `uint32_t`

Default value: `0x10`

**subcluster0.cpu2.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu2.l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: 0x80000

**subcluster0.cpu2.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu2.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu2.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t



Default value: 0x0

**subcluster0.cpu2.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu2.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu2.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu2.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu2.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu2.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster0.cpu2.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

### **`subcluster0.cpu2.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **`subcluster0.cpu2.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **`subcluster0.cpu2.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster0.cpu2.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu2.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu2.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster0.cpu2.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu2.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu2.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu2.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu3.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu3.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu3.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: 0x0

### **subcluster0.cpu3.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter trace\_special\_hlt\_imm16.

Type: bool

Default value: false

### **subcluster0.cpu3.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster0.cpu3.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster0.cpu3.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster0.cpu3.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster0.cpu3.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

### **subcluster0.cpu3.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster0.cpu3.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

### **subcluster0.cpu3.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster0.cpu3.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster0.cpu3.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

### **subcluster0.cpu3.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu3.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu3.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu3.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu3.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu3.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu3.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster0.cpu3.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster0.cpu3.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu3.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu3.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu3.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu3.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu3.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu3.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu3.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu3.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster0.cpu3.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`



**subcluster0.cpu4.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster0.cpu4.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster0.cpu4.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu4.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu4.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu4.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu4.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu4.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu4.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu4.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu4.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu4.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu4.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu4.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu4.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu4.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster0.cpu4.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster0.cpu4.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster0.cpu4.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster0.cpu4.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu4.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu4.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu4.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu4.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu4.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster0.cpu4.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu4.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster0.cpu5.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu5.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu5.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu5.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu5.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu5.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu5.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

#### **subcluster0.cpu5.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu5.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu5.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

#### **subcluster0.cpu5.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu5.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20



**subcluster0.cpu5.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu5.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu5.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu5.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu5.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu5.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**`subcluster0.cpu5.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**`subcluster0.cpu5.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**`subcluster0.cpu5.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**`subcluster0.cpu5.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**`subcluster0.cpu5.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**`subcluster0.cpu5.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**`subcluster0.cpu5.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster0.cpu5.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu5.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu5.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu5.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu6.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu6.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster0.cpu6.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu6.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu6.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu6.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu6.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu6.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster0.cpu6.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu6.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu6.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu6.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu6.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu6.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu6.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu6.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu6.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu6.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster0.cpu6.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster0.cpu6.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu6.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu6.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu6.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu6.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu6.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu6.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu6.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu6.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster0.cpu6.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).



Type: `bool`

Default value: `true`

### **subcluster0.cpu7.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **subcluster0.cpu7.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **subcluster0.cpu7.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **subcluster0.cpu7.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **subcluster0.cpu7.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **subcluster0.cpu7.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu7.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu7.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu7.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu7.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu7.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu7.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu7.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu7.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu7.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu7.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster0.cpu7.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster0.cpu7.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster0.cpu7.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster0.cpu7.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu7.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu7.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu7.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu7.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu7.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster0.cpu7.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu7.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster0.cpu8.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu8.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu8.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu8.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu8.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu8.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu8.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu8.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20



**subcluster0.cpu8.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu8.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu8.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu8.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu8.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu8.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

### **`subcluster0.cpu8.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

### **`subcluster0.cpu8.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **`subcluster0.cpu8.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **`subcluster0.cpu8.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster0.cpu8.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu8.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu8.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster0.cpu8.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu8.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu8.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu8.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu9.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu9.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster0.cpu9.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu9.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu9.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu9.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu9.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu9.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

#### **subcluster0.cpu9.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

#### **subcluster0.cpu9.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu9.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

#### **subcluster0.cpu9.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu9.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu9.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu9.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu9.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu9.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu9.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu9.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu9.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu9.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster0.cpu9.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster0.cpu9.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu9.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu9.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu9.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu9.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster0.cpu9.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu9.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu9.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster0.cpu9.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).



Type: `bool`

Default value: `true`

### **`subcluster0.dcache-hit_latency`**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.dcache-maintenance_latency`**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.dcache-miss_latency`**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.dcache-prefetch_enabled`**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

Type: `bool`

Default value: `false`

### **`subcluster0.dcache-read_access_latency`**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.dcache-size**

L1 D-Cache size in bytes.

Type: `uint32_t`

Default value: `0x10000`

**subcluster0.dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.dcache-write\_access\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.etc.CLAIMTAGS**

Number of claim tags.

Type: `uint8_t`

Default value: 4

**subcluster0.ete.MAX\_INST\_PER\_Q**

Maximum limit for the number of instructions implied by a Q element.

Type: uint16\_t

Default value: 0x1

**subcluster0.ete.NumberOfRSPairs**

Number of resource selector pairs.

Type: uint8\_t

Default value: 8

**subcluster0.ete.PIDR\_CM0D**

TRCPIDR CM0D value.

Type: uint8\_t

Default value: 0

**subcluster0.ete.PIDR\_REVAND**

TRCPIDR REVAND value.

Type: uint8\_t

Default value: 0

**subcluster0.ete.PIDR\_REVISION**

TRCPIDR REVISION value.

Type: uint8\_t

Default value: 0

**subcluster0.ete.Q\_CADENCE**

Number of instruction blocks traced between two Q elements.

Type: uint16\_t

Default value: 0x1

**subcluster0.ete.RES0\_STATEFUL**

Whether RES0 bits are stateful or RAZ/WI.

Type: bool

Default value: `false`

**subcluster0.ete.RETSTACK**

Return stack depth.

Type: `uint8_t`

Default value: 3

**subcluster0.ete.REVISION**

TRCIDR1 revision value.

Type: `uint8_t`

Default value: 0

**subcluster0.ete.SIM\_OVERFLOW\_GRANULARITY**

Number of instruction blocks in each granule, for simulated overflow.

Type: `uint32_t`

Default value: `0x64`

**subcluster0.ete.SIM\_OVERFLOW\_PERCENTAGE**

Percentage of instruction blocks lost in each granule, for simulated overflow.

Type: `uint8_t`

Default value: 0

**subcluster0.ete.SOURCE\_ADDRESS**

Allow generation of source address elements.

Type: `bool`

Default value: `false`

**subcluster0.ete.TRACE\_OUTPUT**

File to which to write trace byte stream.

Type: `string`

Default value: `N/A`

**subcluster0.ete.TRCRSRTA\_FORCED\_EXCEP**

TRCRSR.TA value for a forcibly traced exception.

Type: `bool`

Default value: `false`

### **subcluster0.force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks**

Force MTE tag accesses to **RAZ/WI** and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

Type: `bool`

Default value: `false`

### **subcluster0.force\_zero\_PSTATE\_PAN**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

### **subcluster0.force\_zero\_mpam\_partid\_and\_pmg**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAMO\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

Type: `uint8_t`

Default value: `0`

### **subcluster0.has\_coherent\_icache**

Whether icache invalidation to the point of unification is required for instruction to data coherence.  
true - Invalidate operations not required.

Type: `bool`

Default value: `true`

### **subcluster0.has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

### **subcluster0.has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

### **subcluster0.has\_dot\_product**

Possible values of this parameter are:- 1, feature not implemented.- 2, feature is implemented.

Type: `uint8_t`

Default value: 2

### **subcluster0.has\_ete**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (-plugin or -P).

Type: `bool`

Default value: `false`

### **subcluster0.icache-hit\_latency**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **subcluster0.icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **subcluster0.icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **subcluster0.icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

**subcluster0.icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.icache-size**

L1 I-Cache size in bytes.

Type: uint32\_t

Default value: 0x10000

**subcluster0.invalidate\_code\_cache\_on\_icache\_cmo**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: uint8\_t

Default value: 0

**subcluster0.max\_32bit\_el**

Maximum exception level supporting AArch32 modes. -1: No Support for A32 at any EL, 0: ELO supports A32. This parameter is ignored in Rev0 i.e. when revision\_number = 0.

Type: int8\_t

Default value: 0

**subcluster0.memory\_tagging\_support\_level**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

Type: `uint8_t`

Default value: 3

**subcluster0.ptw\_latency**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

**subcluster0.ras\_pfg\_clock\_mhz**

RAS Pseudo-Fault generation clock rate in MHz.

Type: `uint8_t`

Default value: 12

**subcluster0.revision\_number**

Revision number of TRM implemented by the model. Changing the value can change the model behaviour. Visible as the Variant field in MIDR/MIDR\_EL1. Corresponds to the revision number X in rXpY.

Type: `uint8_t`

Default value: 1

**subcluster0.tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

**subcluster0.tlbi\_stall\_enabled**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: `bool`

Default value: `false`



**subcluster0.treat\_PAC\_as\_NOP**

Non-architecture parameter. Treat Pointer Authentication as **NOP**. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are **NOP**. Branch, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

Type: `bool`

Default value: `false`

**subcluster0.walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.CPUCFR**

Value of CPU Configuration Register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.NUM\_CORES**

Number of cores per cluster.

Type: `uint8_t`

Default value: `1`

**subcluster1.cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**subcluster1.cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**subcluster1.cpu0.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu0.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster1.cpu0.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu0.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu0.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu0.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu0.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu0.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu0.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu0.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu0.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu0.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

**subcluster1.cpu0.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu0.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu0.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu0.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu0.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu0.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu0.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu0.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu0.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu0.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster1.cpu0.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster1.cpu0.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu0.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu0.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu0.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu0.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu0.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu0.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu0.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu0.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu0.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster1.cpu1.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster1.cpu1.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu1.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu1.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu1.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu1.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu1.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu1.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu1.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu1.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`



Default value: 0x0

#### **subcluster1.cpu1.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

#### **subcluster1.cpu1.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu1.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu1.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu1.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu1.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu1.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu1.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu1.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu1.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu1.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster1.cpu1.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster1.cpu1.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster1.cpu1.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster1.cpu1.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu1.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu1.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu1.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu1.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu1.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster1.cpu1.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu1.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster1.cpu2.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu2.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu2.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu2.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter trace\_special\_hlt\_imm16.

Type: `bool`

Default value: `false`

**subcluster1.cpu2.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu2.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

**subcluster1.cpu2.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster1.cpu2.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu2.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu2.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu2.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu2.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu2.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster1.cpu2.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

### **`subcluster1.cpu2.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **`subcluster1.cpu2.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **`subcluster1.cpu2.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster1.cpu2.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu2.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster1.cpu2.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster1.cpu2.semihosting-stack_limit`**

Virtual address of stack limit.



Type: `uint64_t`

Default value: `0xf000000`

#### **`subcluster1.cpu2.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster1.cpu2.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu2.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster1.cpu3.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu3.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu3.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu3.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu3.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu3.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster1.cpu3.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu3.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu3.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

### **`subcluster1.cpu3.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

### **`subcluster1.cpu3.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu3.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu3.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu3.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster1.cpu3.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster1.cpu3.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster1.cpu3.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster1.cpu3.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu3.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: 0x0

**subcluster1.cpu3.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: 0xf000000

**subcluster1.cpu3.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu3.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu3.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu3.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu3.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster1.cpu4.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu4.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster1.cpu4.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu4.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu4.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu4.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu4.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu4.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu4.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu4.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu4.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu4.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

#### **`subcluster1.cpu4.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`



**subcluster1.cpu4.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu4.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu4.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu4.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu4.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu4.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu4.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu4.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu4.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster1.cpu4.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster1.cpu4.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu4.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu4.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu4.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu4.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu4.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster1.cpu4.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu4.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu4.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster1.cpu4.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster1.cpu5.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster1.cpu5.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu5.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu5.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu5.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu5.force-fpsid`**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu5.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu5.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu5.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu5.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu5.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu5.l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: 0x80000

**subcluster1.cpu5.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu5.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu5.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu5.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu5.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu5.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu5.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster1.cpu5.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster1.cpu5.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: 0x123456

**subcluster1.cpu5.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster1.cpu5.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster1.cpu5.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster1.cpu5.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster1.cpu5.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: true

**subcluster1.cpu5.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu5.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu5.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster1.cpu5.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu5.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000



**subcluster1.cpu5.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu5.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster1.cpu6.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu6.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster1.cpu6.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu6.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu6.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu6.force-fpsid`**

Override the FPSID value.

Type: `bool`

Default value: `true`

### **`subcluster1.cpu6.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu6.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu6.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu6.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu6.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to

the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu6.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

#### **`subcluster1.cpu6.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu6.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu6.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu6.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu6.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu6.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu6.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu6.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu6.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu6.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster1.cpu6.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster1.cpu6.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu6.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu6.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu6.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu6.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu6.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu6.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: 0xf000000

**subcluster1.cpu6.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu6.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster1.cpu6.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: true

**subcluster1.cpu7.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: false

**subcluster1.cpu7.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: bool

Default value: false

**subcluster1.cpu7.CRYPTODISABLE**

Disable cryptographic features.

Type: bool

Default value: false

**subcluster1.cpu7.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu7.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu7.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu7.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu7.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu7.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu7.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu7.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu7.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster1.cpu7.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu7.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu7.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.



Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu7.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu7.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

### **`subcluster1.cpu7.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

### **`subcluster1.cpu7.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu7.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu7.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu7.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster1.cpu7.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster1.cpu7.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster1.cpu7.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster1.cpu7.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu7.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: 0x0

**subcluster1.cpu7.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: 0xf000000

**subcluster1.cpu7.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu7.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu7.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu7.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu7.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster1.cpu8.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu8.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster1.cpu8.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu8.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu8.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu8.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu8.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu8.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu8.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu8.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu8.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu8.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

### **`subcluster1.cpu8.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu8.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu8.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu8.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu8.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu8.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu8.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu8.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu8.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu8.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster1.cpu8.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster1.cpu8.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu8.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu8.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu8.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu8.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu8.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster1.cpu8.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu8.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu8.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster1.cpu8.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).



Type: `bool`

Default value: `true`

#### **`subcluster1.cpu9.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu9.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu9.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu9.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu9.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu9.force-fpsid`**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu9.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster1.cpu9.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu9.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu9.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster1.cpu9.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster1.cpu9.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: 0x123456

**subcluster1.cpu9.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster1.cpu9.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster1.cpu9.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster1.cpu9.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster1.cpu9.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: true

**subcluster1.cpu9.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu9.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster1.cpu9.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu9.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu9.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu9.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster1.dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.dcache-maintenance\_latency**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.dcache-miss\_latency**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.dcache-prefetch\_enabled**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

Type: `bool`

Default value: `false`

**subcluster1.dcache-read\_access\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.dcache-size**

L1 D-Cache size in bytes.

Type: uint32\_t

Default value: 0x8000

**subcluster1.dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.dcache-write\_access\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.error\_record\_feature\_register**

RAS feature register values. An array of JSON objects. The JSON schema for the array is:

```
[{"ED":0x0, "IMPDEF_3_2":0x0, "UI":0x0, "FI":0x0, "UE":0x0, "CFI":0x0, "CEC":0x0, "RP":0x0, "DUI":0x0, "CEO":0x0, "CI":0x0, "TS":0x0, "INJ":0x0, "FRX":0x0, "UC":0x0, "UEU":0x0, "UER":0x0, "UEO":0x0, "DE":0x0, "CE":0x0, "Visibility":"Core"},other_feature_register_values].
```

Where ED,UI,FI,CE and UE have valid values between 0x0 - 0x3. CFI and DUI have valid values 0x0, 0x2 and 0x3. CEC has valid values 0x0,0x2 or 0x4. RP,CEO,INJ,FRX,UC,UEU,UER,UEO,DE has valid values 0x0 or 0x1. CI and TS has valid values of 0x0, 0x1 and 0x2. Visibility has valid values "Core" or "Cluster".

Type: string

Default value: "[{"ED":0x2, "IMPDEF\_3\_2":0x1, "UI":0x2, "FI":0x2, "UE":0x1, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x0, "INJ":0x1, "CI":0x2, "TS":0x0, "Visibility":"Cluster"}, {"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x1, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x0, "INJ":0x1, "CI":0x0, "TS":0x0}]",

**subcluster1.ete.CLAIMTAGS**

Number of claim tags.

Type: uint8\_t

Default value: 4

**subcluster1.ete.MAX\_INST\_PER\_Q**

Maximum limit for the number of instructions implied by a Q element.

Type: uint16\_t

Default value: 0x1

**subcluster1.ete.NumberOfRSPairs**

Number of resource selector pairs.

Type: uint8\_t

Default value: 8



**subcluster1.ete.PIDR\_CM0D**

TRCPIDR CM0D value.

Type: uint8\_t

Default value: 0

**subcluster1.ete.PIDR\_REVAND**

TRCPIDR REVAND value.

Type: uint8\_t

Default value: 0

**subcluster1.ete.PIDR\_REVISION**

TRCPIDR REVISION value.

Type: uint8\_t

Default value: 0

**subcluster1.ete.Q\_CADENCE**

Number of instruction blocks traced between two Q elements.

Type: uint16\_t

Default value: 0x1

**subcluster1.ete.RES0\_STATEFUL**

Whether **RES0** bits are stateful or **RAZ/WI**.

Type: bool

Default value: false

**subcluster1.ete.RETSTACK**

Return stack depth.

Type: uint8\_t

Default value: 3

**subcluster1.ete.REVISION**

TRCIDR1 revision value.

Type: uint8\_t

Default value: 1

**subcluster1.ete.SIM\_OVERFLOW\_GRANULARITY**

Number of instruction blocks in each granule, for simulated overflow.

Type: `uint32_t`

Default value: `0x64`

**subcluster1.ete.SIM\_OVERFLOW\_PERCENTAGE**

Percentage of instruction blocks lost in each granule, for simulated overflow.

Type: `uint8_t`

Default value: `0`

**subcluster1.ete.SOURCE\_ADDRESS**

Allow generation of source address elements.

Type: `bool`

Default value: `false`

**subcluster1.ete.TRACE\_OUTPUT**

File to which to write trace byte stream.

Type: `string`

Default value: `N/A`

**subcluster1.ete.TRCSRSTA\_FORCED\_EXCEP**

TRCSRSTA value for a forcibly traced exception.

Type: `bool`

Default value: `false`

**subcluster1.force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks**

Force MTE tag accesses to **RAZ/WI** and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

Type: `bool`

Default value: `false`

**subcluster1.force\_zero\_PSTATE\_PAN**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

**subcluster1.force\_zero\_mpam\_partid\_and\_pmg**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

Type: `uint8_t`

Default value: `0`

**subcluster1.has\_coherent\_icache**

Whether icache invalidation to the point of unification is required for instruction to data coherence. `true` - Invalidate operations not required.

Type: `bool`

Default value: `true`

**subcluster1.has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

**subcluster1.has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

**subcluster1.has\_enhanced\_pan**

Implements Armv8.7 Enhanced PAN feature (FEAT\_PAN3) values of this parameter are:- 1, feature is implemented if ARMv8.7 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: `2`

**subcluster1.has\_ete**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (-plugin or -P).

Type: `bool`

Default value: `false`

**subcluster1.has\_large\_va**

Implement support for the extended 52-bit virtual addresses from ARMv8.2 (FEAT\_LVA). values of this parameter are: - 0, feature is not enabled. - 1, feature is implemented if ARMv8.2 is enabled. - 2, feature is implemented.

Type: `uint8_t`

Default value: 0

**subcluster1.has\_statistical\_profiling**

Whether Statistical Based Profiling is implemented (FEAT\_SPE).

Type: `bool`

Default value: `false`

**subcluster1.icache-hit\_latency**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster1.icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster1.icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster1.icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

Type: bool

Default value: false

**subcluster1.icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.icache-size**

L1 I-Cache size in bytes.

Type: uint32\_t

Default value: 0x8000

**subcluster1.instruction\_tlb\_size**

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

Type: uint32\_t

Default value: 0x0

**subcluster1.invalidate\_code\_cache\_on\_icache\_cmo**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: uint8\_t

Default value: 0

### **subcluster1.memory\_tagging\_support\_level**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

Type: uint8\_t

Default value: 3

### **subcluster1.pmu-num\_counters**

Number of PMU counters implemented.

Type: uint8\_t

Default value: 6

### **subcluster1.pseudo\_fault\_generation\_feature\_register**

ARMv8.4 Standard Pseudo-fault generation feature register values. JSON schema for the parameter value is: [{"OF":false, "UC":false, "UEU":false, "UER":false, "UEO":false, "DE":false, "CE":0x0, "CI":false, "ER":false, "PN":false, "AV":false, "MV":false, "SYN":false, "R":false, "NA":false}, other\_psuedo-fault\_generating\_features\_register\_values]. Where OF, UC, UEU, UER, UEO, DE, CI, ER, PN, AV, MV, SYN, and R have valid false(NOT\_SUPPORTED) and true(FEATURE\_CONTROLLABLE), where CE can have 0(NOT\_SUPPORTED), 1(NONSPECIFIC\_CE\_SUPPORTED) and 3(TRANSIENT\_OR\_PERSISTENT\_CE\_SUPPORTED) and NA can have false(component fakes detection on next access) or true(component fakes detection spontaneously). Effective only when ERXFR's INJ field allows it or has\_ras\_fault\_injection is true.

Type: string

Default value: "[{"OF":true, "UC":true, "UEU":false, "UER":false, "UEO":false, "DE":0x1, "CE":0x1, "CI":true, "ER":false, "PN":true, "AV":false, "MV":true, "SYN":true, "R":true}, {"OF":false, "UC":true, "UEU":false, "UER":false, "UEO":false, "DE":0x1, "CE":0x1, "CI":false, "ER":false, "PN":false, "AV":false, "MV":false, "SYN":false, "R":true}]"

### **subcluster1.ptw\_latency**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

### **subcluster1.stage12\_tlb\_size**

Number of stage1+2 tlb entries. Valid values are 0 or >= 4.

Type: uint32\_t

Default value: 0x80

### **subcluster1.tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

### **subcluster1.tlbi\_stall\_enabled**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: bool

Default value: false

### **subcluster1.treat\_PAC\_as\_NOP**

Non-architecture parameter. Treat Pointer Authentication as **NOP**. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are **NOP**. Brach, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

Type: bool

Default value: false

### **subcluster1.walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

### **subcluster2.CPUCFR**

Value of CPU Configuration Register.

Type: uint64\_t

Default value: 0x0

### **subcluster2.NUM\_CORES**

Number of cores per cluster.

Type: uint8\_t

Default value: 1

**subcluster2.cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**subcluster2.cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**subcluster2.cpu0.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster2.cpu0.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster2.cpu0.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster2.cpu0.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu0.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`



Default value: `false`

### **`subcluster2.cpu0.force-fpsid`**

Override the FPSID value.

Type: `bool`

Default value: `true`

### **`subcluster2.cpu0.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu0.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu0.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu0.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu0.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu0.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x100000`

### **`subcluster2.cpu0.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu0.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu0.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu0.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu0.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

### **`subcluster2.cpu0.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

### **`subcluster2.cpu0.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster2.cpu0.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster2.cpu0.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

### **`subcluster2.cpu0.semihosting-T32_HLT`**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

### **`subcluster2.cpu0.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

### **`subcluster2.cpu0.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**`subcluster2.cpu0.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**`subcluster2.cpu0.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**`subcluster2.cpu0.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**`subcluster2.cpu0.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**`subcluster2.cpu0.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**`subcluster2.cpu0.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu0.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu0.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster2.cpu0.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: true

**subcluster2.cpu1.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: false

**subcluster2.cpu1.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: bool

Default value: false

**subcluster2.cpu1.CRYPTODISABLE**

Disable cryptographic features.

Type: bool

Default value: false

**subcluster2.cpu1.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu1.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster2.cpu1.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster2.cpu1.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu1.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu1.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu1.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu1.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu1.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x100000

**subcluster2.cpu1.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu1.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu1.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu1.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu1.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

### **`subcluster2.cpu1.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

### **`subcluster2.cpu1.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster2.cpu1.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster2.cpu1.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`



**subcluster2.cpu1.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster2.cpu1.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster2.cpu1.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster2.cpu1.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster2.cpu1.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster2.cpu1.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu1.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu1.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster2.cpu1.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu1.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster2.cpu1.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster2.cpu1.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster2.cpu2.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster2.cpu2.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster2.cpu2.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster2.cpu2.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu2.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster2.cpu2.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster2.cpu2.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu2.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu2.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu2.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu2.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu2.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x100000`

### **`subcluster2.cpu2.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu2.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu2.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu2.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu2.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster2.cpu2.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster2.cpu2.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu2.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster2.cpu2.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster2.cpu2.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster2.cpu2.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster2.cpu2.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster2.cpu2.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster2.cpu2.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster2.cpu2.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu2.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster2.cpu2.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster2.cpu2.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster2.cpu2.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu2.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster2.cpu2.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster2.cpu3.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster2.cpu3.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu3.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu3.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu3.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu3.force-fpsid`**

Override the FPSID value.

Type: `bool`

Default value: `true`



**subcluster2.cpu3.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu3.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu3.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu3.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu3.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu3.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x100000

#### **subcluster2.cpu3.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster2.cpu3.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster2.cpu3.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster2.cpu3.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster2.cpu3.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

#### **subcluster2.cpu3.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster2.cpu3.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster2.cpu3.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster2.cpu3.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: 0x123456

**subcluster2.cpu3.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster2.cpu3.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster2.cpu3.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster2.cpu3.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster2.cpu3.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: true

**subcluster2.cpu3.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu3.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster2.cpu3.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster2.cpu3.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster2.cpu3.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu3.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster2.cpu3.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster2.cpu4.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster2.cpu4.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster2.cpu4.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster2.cpu4.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu4.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu4.force-fpsid`**

Override the FPSID value.

Type: `bool`

Default value: `true`

### **`subcluster2.cpu4.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu4.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu4.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu4.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu4.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to

the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster2.cpu4.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x100000`

#### **`subcluster2.cpu4.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster2.cpu4.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster2.cpu4.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster2.cpu4.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu4.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster2.cpu4.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster2.cpu4.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu4.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu4.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster2.cpu4.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster2.cpu4.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t



Default value: 171

**subcluster2.cpu4.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster2.cpu4.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster2.cpu4.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster2.cpu4.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu4.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu4.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster2.cpu4.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: 0xf000000

#### **subcluster2.cpu4.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

#### **subcluster2.cpu4.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

#### **subcluster2.cpu4.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: true

#### **subcluster2.cpu5.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: false

#### **subcluster2.cpu5.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: bool

Default value: false

#### **subcluster2.cpu5.CRYPTODISABLE**

Disable cryptographic features.

Type: bool

Default value: false

**subcluster2.cpu5.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu5.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster2.cpu5.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster2.cpu5.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu5.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu5.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu5.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu5.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu5.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x100000

**subcluster2.cpu5.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu5.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu5.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu5.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu5.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

### **`subcluster2.cpu5.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

### **`subcluster2.cpu5.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster2.cpu5.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster2.cpu5.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster2.cpu5.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster2.cpu5.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster2.cpu5.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster2.cpu5.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster2.cpu5.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster2.cpu5.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: 0x0

**subcluster2.cpu5.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: 0xf000000

**subcluster2.cpu5.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster2.cpu5.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu5.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster2.cpu5.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster2.cpu5.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster2.cpu6.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster2.cpu6.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster2.cpu6.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster2.cpu6.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu6.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster2.cpu6.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster2.cpu6.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu6.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.



Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu6.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu6.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu6.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu6.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x100000`

### **`subcluster2.cpu6.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu6.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu6.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu6.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu6.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster2.cpu6.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster2.cpu6.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu6.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster2.cpu6.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster2.cpu6.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster2.cpu6.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster2.cpu6.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster2.cpu6.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster2.cpu6.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster2.cpu6.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu6.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster2.cpu6.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster2.cpu6.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster2.cpu6.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu6.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster2.cpu6.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster2.cpu7.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster2.cpu7.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu7.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu7.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu7.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu7.force-fpsid`**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster2.cpu7.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu7.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu7.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu7.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu7.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu7.l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: 0x100000

**subcluster2.cpu7.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu7.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu7.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu7.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu7.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster2.cpu7.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster2.cpu7.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster2.cpu7.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster2.cpu7.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: 0x123456

**subcluster2.cpu7.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster2.cpu7.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster2.cpu7.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster2.cpu7.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`



Default value: N/A

**subcluster2.cpu7.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: true

**subcluster2.cpu7.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu7.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster2.cpu7.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster2.cpu7.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster2.cpu7.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu7.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster2.cpu7.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster2.cpu8.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster2.cpu8.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster2.cpu8.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster2.cpu8.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu8.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu8.force-fpsid`**

Override the FPSID value.

Type: `bool`

Default value: `true`

### **`subcluster2.cpu8.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu8.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu8.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu8.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu8.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to

the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster2.cpu8.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x100000`

#### **`subcluster2.cpu8.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster2.cpu8.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster2.cpu8.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster2.cpu8.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu8.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster2.cpu8.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster2.cpu8.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu8.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu8.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster2.cpu8.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster2.cpu8.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster2.cpu8.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster2.cpu8.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster2.cpu8.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster2.cpu8.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu8.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu8.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster2.cpu8.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: 0xf000000

#### **subcluster2.cpu8.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

#### **subcluster2.cpu8.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

#### **subcluster2.cpu8.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: true

#### **subcluster2.cpu9.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: false

#### **subcluster2.cpu9.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: bool

Default value: false

#### **subcluster2.cpu9.CRYPTODISABLE**

Disable cryptographic features.

Type: bool

Default value: false

**subcluster2.cpu9.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu9.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster2.cpu9.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster2.cpu9.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu9.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu9.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`



**subcluster2.cpu9.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu9.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu9.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x100000

**subcluster2.cpu9.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu9.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu9.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu9.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu9.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

### **`subcluster2.cpu9.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

### **`subcluster2.cpu9.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster2.cpu9.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster2.cpu9.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster2.cpu9.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster2.cpu9.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster2.cpu9.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster2.cpu9.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster2.cpu9.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster2.cpu9.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu9.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu9.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster2.cpu9.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu9.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster2.cpu9.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster2.cpu9.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster2.dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.dcache-maintenance\_latency**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.dcache-miss\_latency**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.dcache-prefetch\_enabled**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

Type: bool

Default value: false

**subcluster2.dcache-read\_access\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.dcache-size**

L1 D-Cache size in bytes.

Type: uint32\_t

Default value: 0x8000

### **subcluster2.dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster2.dcache-write\_access\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster2.dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster2.error\_record\_feature\_register**

RAS feature register values. An array of JSON objects. The JSON schema for the array is: [{"ED":0x0, "IMPDEF\_3\_2":0x0, "UI":0x0, "FI":0x0, "UE":0x0, "CFI":0x0, "CEC":0x0, "RP":0x0, "DUI":0x0, "CEO":0x0, "CI":0x0, "TS":0x0, "INJ":0x0, "FRX":0x0, "UC":0x0, "UEU":0x0, "UER":0x0, "UEO":0x0, "DE":0x0, "CE":0x0, "Visibility":"Core"},other\_feature\_register\_values]. Where ED,UI,FI,CE and UE have valid values between 0x0 - 0x3. CFI and DUI have valid values 0x0, 0x2 and 0x3. CEC has valid values 0x0,0x2 or 0x4. RP,CEO,INJ,FRX,UC,UEU,UER,UEO,DE has valid values 0x0 or 0x1. CI and TS has valid values of 0x0, 0x1 and 0x2. Visibility has valid values "Core" or "Cluster".

Type: string

Default value: "[{"ED":0x2, "IMPDEF\_3\_2":0x0, "UI":0x2, "FI":0x2, "UE":0x1, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x0, "INJ":0x1, "CI":0x0, "TS":0x0, "Visibility":"Cluster"}, {"ED":0x2, "IMPDEF\_3\_2":0x0, "UI":0x2, "FI":0x2, "UE":0x1, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x0, "INJ":0x1, "CI":0x0, "TS":0x0}]",

**subcluster2.ete.CLAIMTAGS**

Number of claim tags.

Type: uint8\_t

Default value: 32

**subcluster2.ete.MAX\_INST\_PER\_Q**

Maximum limit for the number of instructions implied by a Q element.

Type: uint16\_t

Default value: 0x1

**subcluster2.ete.NumberOfRSPairs**

Number of resource selector pairs.

Type: uint8\_t

Default value: 8

**subcluster2.ete.PIDR\_CM0D**

TRCPIDR CM0D value.

Type: uint8\_t

Default value: 0

**subcluster2.ete.PIDR\_REVAND**

TRCPIDR REVAND value.

Type: uint8\_t

Default value: 0

**subcluster2.ete.PIDR\_REVISION**

TRCPIDR REVISION value.

Type: uint8\_t

Default value: 0

**subcluster2.ete.Q\_CADENCE**

Number of instruction blocks traced between two Q elements.

Type: uint16\_t

Default value: 0x1

**subcluster2.ete.RES0\_STATEFUL**

Whether **RES0** bits are stateful or **RAZ/WI**.

Type: `bool`

Default value: `false`

**subcluster2.ete.RETSTACK**

Return stack depth.

Type: `uint8_t`

Default value: `3`

**subcluster2.ete.REVISION**

TRCIDR1 revision value.

Type: `uint8_t`

Default value: `0`

**subcluster2.ete.SIM\_OVERFLOW\_GRANULARITY**

Number of instruction blocks in each granule, for simulated overflow.

Type: `uint32_t`

Default value: `0x64`

**subcluster2.ete.SIM\_OVERFLOW\_PERCENTAGE**

Percentage of instruction blocks lost in each granule, for simulated overflow.

Type: `uint8_t`

Default value: `0`

**subcluster2.ete.SOURCE\_ADDRESS**

Allow generation of source address elements.

Type: `bool`

Default value: `false`

**subcluster2.ete.TRACE\_OUTPUT**

File to which to write trace byte stream.

Type: `string`

Default value: `N/A`



**subcluster2.ete.TRCRSRTA\_FORCED\_EXCEP**

TRCRSR.TA value for a forcibly traced exception.

Type: `bool`

Default value: `false`

**subcluster2.ext\_abort\_normal\_noncacheable\_read\_is\_sync**

Synchronous reporting of normal noncacheable-read external aborts.

Type: `bool`

Default value: `true`

**subcluster2.ext\_abort\_so\_write\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `uint8_t`

Default value: 2

**subcluster2.force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks**

Force MTE tag accesses to **RAZ/WI** and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

Type: `bool`

Default value: `false`

**subcluster2.force\_zero\_PSTATE\_PAN**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

**subcluster2.force\_zero\_mpam\_partid\_and\_pmg**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

Type: `uint8_t`

Default value: 0

**subcluster2.has\_coherent\_icache**

Whether icache invalidation to the point of unification is required for instruction to data coherence.  
true - Invalidate operations not required.

Type: `bool`

Default value: `true`

**subcluster2.has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

**subcluster2.has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

**subcluster2.has\_enhanced\_pan**

Implements Armv8.7 Enhanced PAN feature (FEAT\_PAN3) values of this parameter are:- 1, feature is implemented if ARMv8.7 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 2

**subcluster2.has\_ete**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (-plugin or -P).

Type: `bool`

Default value: `false`

**subcluster2.has\_large\_va**

Implement support for the extended 52-bit virtual addresses from ARMv8.2 (FEAT\_LVA). values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.2 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

**subcluster2.icache-hit\_latency**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

Type: bool

Default value: false

**subcluster2.icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to

the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster2.icache-size`**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: `0x8000`

#### **`subcluster2.instruction_tlb_size`**

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

Type: `uint32_t`

Default value: `0x0`

#### **`subcluster2.invalidate_code_cache_on_icache_cmo`**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: `uint8_t`

Default value: `0`

#### **`subcluster2.memory_tagging_support_level`**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

Type: `uint8_t`

Default value: `3`

#### **`subcluster2.pmu-num_counters`**

Number of PMU counters implemented.

Type: `uint8_t`

Default value: `6`

#### **`subcluster2.pseudo_fault_generation_feature_register`**

ARMv8.4 Standard Pseudo-fault generation feature register values. JSON schema for the parameter value is: `[{"OF":false, "UC":false, "UEU":false, "UER":false, "UEO":false, "DE":false,`

"CE":0x0, "CI":false, "ER":false, "PN":false, "AV":false, "MV":false, "SYN":false, "R":false, "NA":false}, other\_psuedo-fault\_generating\_features\_register\_values]. Where OF, UC, UEU, UER, UEO, DE, CI, ER, PN, AV, MV, SYN, and R have valid false(NOT\_SUPPORTED) and true(FEATURE\_CONTROLLABLE), where CE can have 0(NOT\_SUPPORTED), 1(NONSPECIFIC\_CE\_SUPPORTED) and 3(TRANSIENT\_OR\_PERSISTENT\_CE\_SUPPORTED) and NA can have false(component fakes detection on next access) or true(component fakes detection spontaneously). Effective only when ERXFR's INJ field allows it or has\_ras\_fault\_injection is true.

Type: string

Default value: "[{"OF":true, "UC":true, "UEU":false, "UER":false, "UEO":false, "DE":0x1, "CE":0x1, "CI":true, "ER":false, "PN":true, "AV":false, "MV":true, "SYN":true, "R":true}, {"OF":false, "UC":true, "UEU":false, "UER":false, "UEO":false, "DE":0x1, "CE":0x1, "CI":false, "ER":false, "PN":false, "AV":false, "MV":false, "SYN":false, "R":true}]"

### **subcluster2.ptw\_latency**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

### **subcluster2.stage12\_tlb\_size**

Number of stage1+2 tlb entries. Valid values are 0 or >= 4.

Type: uint32\_t

Default value: 0x80

### **subcluster2.tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

### **subcluster2.tlbi\_stall\_enabled**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: bool

Default value: false

**subcluster2.trace\_physical\_registers\_when\_host\_virtualisation\_enabled**

Trace sysreg accesses to physical register accessed (0=disabled, 1=Trace only ELR/SPSR\_EL1 as ELR/SPSR\_EL2, 2=Trace all redirected registers as physical registers affecting all features inc. the host virtualised registers.

Type: `uint8_t`

Default value: 1

**subcluster2.treat\_PAC\_as\_NOP**

Non-architecture parameter. Treat Pointer Authentication as **NOP**. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are **NOP**. Branch, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

Type: `bool`

Default value: `false`

**subcluster2.walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

## 3.49 ARMCortexA520AECT

Defined in `LISA/ARMCortexA520AECT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following parameters were added:

- `has_delayed_dbgreg`
- `has_delayed_sysreg`

### AE-specific features implemented

The only AE-specific features implemented in the model are Split-Lock and Lock-Step, with the following limitations:

- The implementation of Split-Lock and Lock-Step in the model is simply a halving of the number of cores available.
- Lock-step is implemented by a parameter that can be true or false for the entire execution only, rather than by a signal that can be enabled during execution.

As the model does not implement true Split-Lock functionality, the ports and parameters are contiguous. For example, for a 4-core CPU with Split-Lock enabled, `cpu0` and `cpu1` identify the available cores and associated ports, not `cpu0` and `cpu2`.

- Hybrid mode is not modeled in the DSU.

## Iris and MTI instances for ARMCortexA520AECT

This model has the following Iris instances:

Name	Instance type
ARMCortexA520AECT	Cluster_ARM_Cortex-A520AE
ARMCortexA520AECT.AMU	PVBusLogger
ARMCortexA520AECT.AMU.mapper	PVBusMapper
ARMCortexA520AECT.DAP	PVBusLogger
ARMCortexA520AECT.DAP.mapper	PVBusMapper
ARMCortexA520AECT.DSU	DSU-120
ARMCortexA520AECT.DSU.PPU_cluster	PPUv1
ARMCortexA520AECT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA520AECT.DSU.PPU_core0	PPUv1
ARMCortexA520AECT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexA520AECT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA520AECT.DSU.mpam_busslave	PVBusSlave
ARMCortexA520AECT.DSU.shared_cache	PVCache
ARMCortexA520AECT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA520AECT.DSU.shared_cache.upstream[Y] (where Y = 0-4)	PVBusSlave
ARMCortexA520AECT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA520AECT.MMAP	PVBusLogger
ARMCortexA520AECT.MMAP.mapper	PVBusMapper
ARMCortexA520AECT.RAS	PVBusLogger
ARMCortexA520AECT.RAS.mapper	PVBusMapper
ARMCortexA520AECT.cpu0	ARM_Cortex-A520AE
ARMCortexA520AECT.cpu0.UTLB	TLB
ARMCortexA520AECT.cpu0.debug_rom	debug_rom
ARMCortexA520AECT.cpu0.dtlb	TLB
ARMCortexA520AECT.cpu0.l1dcache	PVCache
ARMCortexA520AECT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA520AECT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA520AECT.cpu0.l1icache	PVCache
ARMCortexA520AECT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster

Name	Instance type
ARMCortexA520AECT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA520AECT.cpu0.l2cache	PVCache
ARMCortexA520AECT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA520AECT.cpu0.l2cache.upstream[U] (where U = 0-1)	PVBusSlave
ARMCortexA520AECT.ext_bus	PVBusLogger
ARMCortexA520AECT.ext_bus.mapper	PVBusMapper
ARMCortexA520AECT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA520AECT.global_debug_rom	debug_rom
ARMCortexA520AECT.secondary_debug_rom	debug_rom
ARMCortexA520AECT.sve	ScalableVectorExtension

This model has the following MTI trace components:

Name	Component type
ARMCortexA520AECT.AMU	PVBusLogger
ARMCortexA520AECT.AMU.mapper	PVBusMapper
ARMCortexA520AECT.DAP	PVBusLogger
ARMCortexA520AECT.DAP.mapper	PVBusMapper
ARMCortexA520AECT.DSU	DSU-120
ARMCortexA520AECT.DSU.PPU_cluster	PPUv1
ARMCortexA520AECT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA520AECT.DSU.PPU_core0	PPUv1
ARMCortexA520AECT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexA520AECT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA520AECT.DSU.mpam_busslave	PVBusSlave
ARMCortexA520AECT.DSU.shared_cache	PVCache
ARMCortexA520AECT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA520AECT.DSU.shared_cache.upstream[Y] (where Y = 0-4)	PVBusSlave
ARMCortexA520AECT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA520AECT.MMAP	PVBusLogger
ARMCortexA520AECT.MMAP.mapper	PVBusMapper
ARMCortexA520AECT.RAS	PVBusLogger
ARMCortexA520AECT.RAS.mapper	PVBusMapper
ARMCortexA520AECT.cpu0	ARM_Cortex-A520AE
ARMCortexA520AECT.cpu0.UTLB	TLB
ARMCortexA520AECT.cpu0.l1dcache	PVCache
ARMCortexA520AECT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA520AECT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA520AECT.cpu0.l1licache	PVCache
ARMCortexA520AECT.cpu0.l1licache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA520AECT.cpu0.l1licache.upstream[0]	PVBusSlave



Name	Component type
ARMCortexA520AECT.cpu0.l2cache	PVCache
ARMCortexA520AECT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA520AECT.cpu0.l2cache.upstream[U] (where U = 0-1)	PVBusSlave
ARMCortexA520AECT.ext_bus	PVBusLogger
ARMCortexA520AECT.ext_bus.mapper	PVBusMapper
ARMCortexA520AECT.gic_cpuif_decoder_cluster	GIcV3CPUInterfaceDecoder

## Ports for ARMCortexA520AECT

Port	Direction	Protocol	Description
acp_s	slave	PVBus	AXI ACP slave port.
AEND0MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND2MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND3MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART2MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	slave	Signal	CHI defined pins.
broadcastcachemaint	slave	Signal	ACE defined pins.
broadcastouter	slave	Signal	ACE defined pins.
broadcastpersist	slave	Signal	CHI defined pins.
cfgend	slave	Signal	This signal is for EE bit initialisation.
cfgsdisable	slave	Signal	This signal disables write access to some secure Interrupt Controller registers.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcs_m_pchannel	master	PChannel	Cluster PCSM signal
cluster_powerdown_out	master	Signal	DEPRECATED - An output from PPUs that informs thermal controller of the cluster power info
cluster_ppu_hw_stat	master	Value	Cluster PPU power state

Port	Direction	Protocol	Description
clusterid	slave	Value	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set ManagerID64 of the core.
clusterpmuirq	master	Signal	DynamiQ pmu irq
CNTHPIRQ	master	Signal	Timer signals to SOC.
CNTHPSIRQ	master	Signal	Timer signals to SOC
CNTHVIRQ	master	Signal	Timer signals to SOC.
CNTHVSIQ	master	Signal	Timer signals to SOC
CNTPNSIRQ	master	Signal	Timer signals to SOC.
CNTPSIQ	master	Signal	Timer signals to SOC.
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.
CNTVIRQ	master	Signal	Timer signals to SOC.
commirq	master	Signal	Interrupt signal from debug communications channel.
core_clk_in	slave	ClockSignal	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
core_pcsn_pchannel	master	PChannel	Core PCSM signals
core_powerdown_out	master	Signal	DEPRECATED - An output from PPUs that informs thermal controller of the core power info
core_ppu_hw_stat	master	Value	Notify the power state change inside the PPU.
coreerrirq	master	Signal	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq	master	Signal	Fault indicator for a detected 1 or 2 bit ECC error
coreinstrrun	master	Signal	Core Running State
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
ctidbgirq	master	Signal	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	slave	Signal	External debug interface.
dbgnopwrdown	master	Signal	No power-down request.
dbgprupreq	master	Signal	Debug power up request.
defaultMP	slave	Signal	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main master interface ports.
dev_debug_s	slave	PVBus	External debug interface.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.

Port	Direction	Protocol	Description
gicreset	master	Signal	An output from PPU's that can be used to reset GICCLK domain components
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	slave	PVBus	External debug interface.
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	master	Signal	PPU cluster interrupt.
ppu_cluster_isolate	master	Signal	PPU Cluster Domain isolation control.
ppu_cluster_wakerequest	slave	Signal	PPU cluster wake request signal.
ppu_core_irq	master	Signal	PPU core interrupt.
ppu_core_isolate	master	Signal	PPU Core Domain isolation control.
ppu_core_wakerequest	slave	Signal	PPU core wake request signal.
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	master	PVBus	The core will generate bus requests on this port.
pvbus_periph_m	master	PVBus	The core can generate peripheral bus request on this port.
rei	slave	Signal	Per core RAM Error Interrupt.
reset	slave	Signal	Reset signal to cluster.
rvbaraddr	slave	Value_64	Reset vector base address.
sci_m	master	SystemCoherencyInterface	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei	slave	Signal	Per core System Error physical pins.
spiden	slave	Signal	External debug interface.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq	master	Signal	Interrupt signal from the trace buffer unit.
utility_bus_s	slave	PVBus	Utility bus slave
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfiq	slave	Signal	Virtualised FIQ.
virq	slave	Signal	Virtualised IRQ.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Per core virtual System Error physical pins.

## Parameters for ARMCortexA520AECT

### cpuX.CFGEND

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`cpuX.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`cpuX.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`cpuX.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**cpuX.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**cpuX.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**cpuX.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **cpuX.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

### **cpuX.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **cpuX.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

### **cpuX.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **cpuX.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

### **cpuX.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**cpuX.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**cpuX.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**cpuX.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**cpuX.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**cpuX.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**cpuX.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**cpuX.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**cpuX.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**cpuX.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**cpuX.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**cpuX.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**cpuX.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**cpuX.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**cpuX.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.



Type: `bool`

Default value: `false`

### **`cpuX.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`AEND0_DEFAULT`**

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

Type: `uint64_t`

Default value: `0x0`

### **`AEND1_DEFAULT`**

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

Type: `uint64_t`

Default value: `0x0`

### **`AEND2_DEFAULT`**

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

Type: `uint64_t`

Default value: `0x0`

### **`AEND3_DEFAULT`**

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

Type: `uint64_t`

Default value: `0x0`

### **`ASTART0_DEFAULT`**

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

Type: `uint64_t`

Default value: `0x0`

### **ASTART1\_DEFAULT**

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

Type: `uint64_t`

Default value: `0x0`

### **ASTART2\_DEFAULT**

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

Type: `uint64_t`

Default value: `0x0`

### **ASTART3\_DEFAULT**

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

Type: `uint64_t`

Default value: `0x0`

### **BROADCASTATOMIC**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`

Default value: `true`

### **BROADCASTCACHEMAIN**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

Type: `bool`

Default value: `false`

### **BROADCASTOUTER**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

Type: `bool`

Default value: `false`

### **BROADCASTPERSIST**

Enable broadcasting of cache clean to the point of persistence operations. The `broadcastpersist` signal will override this value if used.

Type: `bool`

Default value: `true`

### **CLUSTER\_ID**

The Cluster ID maps to the affinity levels of the `MPIDR_EL1` register and is evaluated based on the `MPIDR_EL1` layout. If `MPIDR_EL1` supports 16-bit cluster affinity levels, bits [15:8] map to `IDRAFF3`, while bits [7:0] map to `IDRAFF2`. If `MPIDR_EL1` supports 24-bit cluster affinity levels, the bits [23:16] map to `IDRAFF3`, bits [15:8] map to `IDRAFF2`, and bits [7:0] map to `IDRAFF1`. This configuration also updates all relevant component `DEVAFF` registers and is used to set the `ManagerID64` of the core.

Type: `uint32_t`

Default value: `0x0`

### **CMO\_broadcast\_when\_cache\_state\_modelling\_disabled**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. This value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain events even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required.

Type: `uint8_t`

Default value: `1`

### **CPUCFR**

Value of CPU Configuration Register.

Type: `uint64_t`

Default value: `0x0`

### **DBGROMADDR**

Initialization value of `DBGDRAR` register. Bits[55:12] of this register specify the ROM table physical address.

Type: `uint64_t`

Default value: 0x0

**DBGROMADDRV**

If true, set bits[1:0] of the CP15 DBGDRAR to indicate that the address is valid.

Type: bool

Default value: false

**GICDISABLE**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: bool

Default value: true

**NUM\_CORES**

Number of cores per cluster.

Type: uint8\_t

Default value: 1

**bus\_type**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

Type: uint8\_t

Default value: 0

**cluster\_split\_lock\_config**

Default SPLIT/LOCKED config. Directly maps to values of CLUSTERSLCFR. The valid values are: 1 - Only LOCKED configuration support 4 - Only SPLIT configuration support, 5 - Mixed Configuration support. Modes are not software visible, and not modeled. Valid only when enable\_ae\_features is true.

Type: uint8\_t

Default value: 1

**core\_cache\_protection**

core\_cache\_protection can change ERROFR, ERROPFGF and ERROPFGCTL fields. Possible values are: -1: Not implemented (by default), 0: Disabled, 1: Enabled.

Type: int8\_t

Default value: 1

**core\_complex\_mapping**

Defines Complex descriptions for platforms that support several Cores per Complex like Cortex-A510. JSON format: {"complex0": { "cores" : [ 0, 1 ], "l2-cache" : {"exists":1, "size":16MB}}, ... , "complexN": { "cores" : [<core\_list>], "l2-cache" : {"exists":1, "size":16MB}}} where <core\_list> is the list of cores in the complexN. Effective only when the parameter value is not empty.

Type: string

Default value: "{ \"complex0\": { \"cores\": [0, 1], \"l2-cache\" : { \"exists\":1, \"size\": \"16MB\" } }, \"complex1\": { \"cores\": [2, 3], \"l2-cache\" : { \"exists\":1, \"size\": \"16MB\" } }, \"complex2\": { \"cores\": [4, 5], \"l2-cache\" : { \"exists\":1, \"size\": \"16MB\" } }, \"complex3\": { \"cores\": [6, 7], \"l2-cache\" : { \"exists\":1, \"size\": \"16MB\" } }, \"complex4\": { \"cores\": [8, 9], \"l2-cache\" : { \"exists\":1, \"size\": \"16MB\" } }, \"complex5\": { \"cores\": [10, 11], \"l2-cache\" : { \"exists\":1, \"size\": \"16MB\" } }, \"complex6\": { \"cores\": [12, 13], \"l2-cache\" : { \"exists\":1, \"size\": \"16MB\" } } }

**core\_power\_on\_by\_default**

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPUs which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

Type: bool

Default value: false

**cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: uint32\_t

Default value: 0x1

**cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: uint32\_t

Default value: 0x1

**dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**dcache-maintenance\_latency**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-miss\_latency**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-prefetch\_enabled**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

Type: `bool`

Default value: `false`

**dcache-read\_access\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-size**

L1 D-Cache size in bytes.

Type: `uint32_t`

Default value: 0x8000

### **dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **dcache-state\_modelled**

Set whether D-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **dcache-write\_access\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **enable\_lock\_step**

Whether the core is configured in Dual Core Lock Step mode (FEAT\_DCLS).

Type: `bool`

Default value: `false`

### **enable\_simulation\_performance\_optimizations**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of `stage12_tlb_size` parameter to 1024).

Type: `bool`

Default value: `true`

### **`ete.CLAIMTAGS`**

Number of claim tags.

Type: `uint8_t`

Default value: 4

### **`ete.MAX_INST_PER_Q`**

Maximum limit for the number of instructions implied by a Q element.

Type: `uint16_t`

Default value: `0x1`

### **`ete.PIDR_CM0D`**

TRCPIDR CM0D value.

Type: `uint8_t`

Default value: 0

### **`ete.PIDR_REVAND`**

TRCPIDR REVAND value.

Type: `uint8_t`

Default value: 0

### **`ete.PIDR_REVISION`**

TRCPIDR REVISION value.

Type: `uint8_t`

Default value: 0

### **`ete.Q_CADENCE`**

Number of instruction blocks traced between two Q elements.

Type: `uint16_t`

Default value: `0x1`

### **`ete.RES0_STATEFUL`**

Whether **RES0** bits are stateful or **RAZ/WI**.



Type: `bool`

Default value: `false`

#### **`ete.RETSTACK`**

Return stack depth.

Type: `uint8_t`

Default value: 3

#### **`ete.REVISION`**

TRCIDR1 revision value.

Type: `uint8_t`

Default value: 0

#### **`ete.SIM_OVERFLOW_GRANULARITY`**

Number of instruction blocks in each granule, for simulated overflow.

Type: `uint32_t`

Default value: `0x64`

#### **`ete.SIM_OVERFLOW_PERCENTAGE`**

Percentage of instruction blocks lost in each granule, for simulated overflow.

Type: `uint8_t`

Default value: 0

#### **`ete.SOURCE_ADDRESS`**

Allow generation of source address elements.

Type: `bool`

Default value: `false`

#### **`ete.TRACE_OUTPUT`**

File to which to write trace byte stream.

Type: `string`

Default value: `N/A`

#### **`ete.TRCSRSTA_FORCED_EXCEP`**

TRCSRSTA value for a forcibly traced exception.

Type: `bool`

Default value: `false`

### **`force_mte_tag_access_razwi_and_ignore_tag_checks`**

Force MTE tag accesses to **RAZ/WI** and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

Type: `bool`

Default value: `false`

### **`force_zero_PSTATE_PAN`**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

### **`force_zero_mpam_partid_and_pmg`**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAMO\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

Type: `uint8_t`

Default value: `0`

### **`has_actlr2`**

If true ACLTR2 exists and ACTLR2(NS) is aliased to ACTLR\_EL1[63:32].

Type: `bool`

Default value: `true`

### **`has_coherent_icache`**

Whether icache invalidation to the point of unification is required for instruction to data coherence. true - Invalidate operations not required.

Type: `bool`

Default value: `false`

### **`has_delayed_dbgreg`**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

### **has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

### **has\_dot\_product**

Possible values of this parameter are:- 1, feature not implemented.- 2, feature is implemented.

Type: `uint8_t`

Default value: 2

### **has\_ete**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (-plugin or -P).

Type: `bool`

Default value: `false`

### **has\_impdef\_transient\_fault\_protection**

Support the Transient Fault Protection (TFP) flop parity errors through RAS registers (FEAT\_TFP).

Type: `bool`

Default value: `true`

### **has\_peripheral\_port**

If true, an additional AXI peripheral port is configured.

Type: `bool`

Default value: `false`

### **icache-hit\_latency**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

**icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-size**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: `0x8000`

### **`icache-state_modelled`**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **`invalidate_code_cache_on_icache_cmo`**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: `uint8_t`

Default value: `0`

### **`l3cache-has_mpam`**

L3 Cache has MPAM support.

Type: `bool`

Default value: `true`

### **`l3cache-hit_latency`**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-maintenance_latency`**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-miss_latency`**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **l3cache-read\_access\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size l3cache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **l3cache-read\_bus\_width\_in\_bytes**

L3 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

### **l3cache-read\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. l3cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **l3cache-size**

L3 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

### **l3cache-snoop\_data\_transfer\_latency**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **l3cache-snoop\_issue\_latency**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

### **l3cache-ways**

L3 Cache number of ways (sets are implicit from size).

Type: `uint16_t`

Default value: `16`

### **l3cache-write\_access\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size `l3cache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **l3cache-write\_bus\_width\_in\_bytes**

L3 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: `uint32_t`

Default value: `0x10`

### **l3cache-write\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l3cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **memory\_tagging\_support\_level**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

Type: `uint8_t`

Default value: `3`

**mpmm\_accumulator\_multiplier**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator.value of n means the accumulator will use (n \* accumulator value) to calculate the mpmm threshold (MPMM).is provided as a fast model workaround to handle cases where execution of quantum in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

Type: uint8\_t

Default value: 1

**num\_acp**

Number of ACP ports.

Type: uint8\_t

Default value: 0

**num\_nodes**

Number of transport nodes. Zero implies direct-connect configuration.

Type: uint8\_t

Default value: 1

**pmu-num\_counters**

Number of PMU counters implemented.

Type: uint8\_t

Default value: 6

**ptw\_latency**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

**ras\_extra\_configurations**

Miscellaneous configurations for error records. An array of JSON objects. Note for ERXCTLR\_EL1 register it only allows to define the mask value for the IMPDEF fields, ie bits [63:32] and bit 1, but its reset value applies on all fields. Note for ERXMISCN masks - these are 64 bit masks covering the 64 bit registers ERXMISCN\_EL1. E.g. [{"Index": 0, "ERXMISCO\_mask": 0x0, "ERXMISCO\_reset": 0x0, "ERXMISC1\_mask": 0x0, "ERXMISC1\_reset": 0x0, "ERXMISC2\_mask": 0x0, "ERXMISC2\_reset": 0x0, "ERXMISC3\_mask": 0x0, "ERXMISC3\_reset": 0x0, "ERXCTLR\_EL1\_mask": 0x0,



"ERXCTLR\_EL1\_reset": 0x0, {"Index": 1, "ERXMISCO\_mask": 0x0, "ERXMISCO\_reset": 0x0, "ERXSTATUS\_IERR\_mask": 0x300}}].

Type: string

Default value: "[ { \"Index\": 1, \"ERXMISCO\_mask\": 0xFFFFc0003fc3, \"ERXMISC1\_mask\": 0x03F870003FF30f07, \"ERXPPFGCTL\_reset\": 0x1000 }, { \"Index\": 2, \"ERXMISCO\_mask\": 0xFFFFe007ffc0, \"ERXMISCO\_reset\": 0x2, \"ERXSTATUS\_IERR\_mask\": 0x300 , \"ERXMISC1\_mask\": 0x0FF8700ffff31f0f, \"ERXPPFGCTL\_reset\": 0x1000 } ]"

### **ras\_pfg\_clock\_mhz**

RAS Pseudo-Fault generation clock rate in MHz.

Type: uint8\_t

Default value: 12

### **reported\_patch\_level**

Purely cosmetic patch level value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: int8\_t

Default value: -1

### **reported\_revision\_number**

Purely cosmetic revision number value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: int8\_t

Default value: -1

### **tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

### **tlbi\_stall\_enabled**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: bool

Default value: false

**treat\_PAC\_as\_NOP**

Non-architecture parameter. Treat Pointer Authentication as **NOP**. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are **NOP**. Branch, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

Type: `bool`

Default value: `false`

**walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

## 3.50 ARMCortexA520CT

Defined in `LISA/ARMCortexA520CT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p1	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following parameters were added:

- `has_delayed_dbgreg`
- `has_delayed_sysreg`

### About ARMCortexA520CT

The model supports the following features:

- L2 cache is supported at the per-core level only and there is no implementation of Core-Complex with shared L2 cache yet.
- Internal PPU support is present.
- A P-Channel for the cluster and for each core.
- `BROADCASTPERSIST` pin.
- Optional peripheral port.
- Memory-mapped register access to MPAM.

- Per-core clock.
- Utility bus.

Support for the following features is planned for a future release:

- DSU-120 system features are not fully implemented.
- Core-Complex.
- BROADCASTCACHEMAINTPOU pin
- COREINSTRRET and COREINSTRRUN signals.
- Each thread has its own L1Cache and L2Cache instead of a common cache that is shared by all threads of the core.
- Per-thread parameters are not supported but signals are implemented.

The following features will not be implemented:

- DynamIQ features that are irrelevant to the programmers' view simulation, for example:
- Automatic CPU retention mode.
- Level-3 Cache RAM retention.
- 256-bit wide output transactions.
- ECC and parity schemes are hardware-specific so are not supported.
- Self-test features (MBIST).
- Latency configuration.
- Snoop filtering.
- Cache stashing capability.

This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (crypto.dll or crypto.so) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

## Iris and MTI instances for ARM CortexA520CT

This model has the following Iris instances:

Name	Instance type
ARM CortexA520CT	Cluster_ARM_Cortex-A520
ARM CortexA520CT.AMU	PVBusLogger
ARM CortexA520CT.AMU.mapper	PVBusMapper
ARM CortexA520CT.DAP	PVBusLogger
ARM CortexA520CT.DAP.mapper	PVBusMapper
ARM CortexA520CT.DSU	DSU-120
ARM CortexA520CT.DSU.PPU_cluster	PPUv1
ARM CortexA520CT.DSU.PPU_cluster.busslave	PVBusSlave
ARM CortexA520CT.DSU.PPU_core0	PPUv1

Name	Instance type
ARMCortexA520CT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexA520CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA520CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA520CT.DSU.shared_cache	PVCache
ARMCortexA520CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA520CT.DSU.shared_cache.upstream[Y] (where Y = 0–4)	PVBusSlave
ARMCortexA520CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA520CT.MMAP	PVBusLogger
ARMCortexA520CT.MMAP.mapper	PVBusMapper
ARMCortexA520CT.RAS	PVBusLogger
ARMCortexA520CT.RAS.mapper	PVBusMapper
ARMCortexA520CT.cpu0	ARM_Cortex-A520
ARMCortexA520CT.cpu0.UTLB	TLB
ARMCortexA520CT.cpu0.debug_rom	debug_rom
ARMCortexA520CT.cpu0.dtlb	TLB
ARMCortexA520CT.cpu0.l1dcache	PVCache
ARMCortexA520CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA520CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA520CT.cpu0.l1icache	PVCache
ARMCortexA520CT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA520CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA520CT.cpu0.l2cache	PVCache
ARMCortexA520CT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA520CT.cpu0.l2cache.upstream[U] (where U = 0–1)	PVBusSlave
ARMCortexA520CT.ext_bus	PVBusLogger
ARMCortexA520CT.ext_bus.mapper	PVBusMapper
ARMCortexA520CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA520CT.global_debug_rom	debug_rom
ARMCortexA520CT.secondary_debug_rom	debug_rom
ARMCortexA520CT.sve	ScalableVectorExtension

This model has the following MTI trace components:

Name	Component type
ARMCortexA520CT.AMU	PVBusLogger
ARMCortexA520CT.AMU.mapper	PVBusMapper
ARMCortexA520CT.DAP	PVBusLogger
ARMCortexA520CT.DAP.mapper	PVBusMapper
ARMCortexA520CT.DSU	DSU-120
ARMCortexA520CT.DSU.PPU_cluster	PPUv1
ARMCortexA520CT.DSU.PPU_cluster.busslave	PVBusSlave

Name	Component type
ARMCortexA520CT.DSU.PPU_core0	PPUv1
ARMCortexA520CT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexA520CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA520CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA520CT.DSU.shared_cache	PVCache
ARMCortexA520CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA520CT.DSU.shared_cache.upstream[Y] (where Y = 0-4)	PVBusSlave
ARMCortexA520CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA520CT.MMAP	PVBusLogger
ARMCortexA520CT.MMAP.mapper	PVBusMapper
ARMCortexA520CT.RAS	PVBusLogger
ARMCortexA520CT.RAS.mapper	PVBusMapper
ARMCortexA520CT.cpu0	ARM_Cortex-A520
ARMCortexA520CT.cpu0.UTLB	TLB
ARMCortexA520CT.cpu0.l1dcache	PVCache
ARMCortexA520CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA520CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA520CT.cpu0.l1icache	PVCache
ARMCortexA520CT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA520CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA520CT.cpu0.l2cache	PVCache
ARMCortexA520CT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA520CT.cpu0.l2cache.upstream[U] (where U = 0-1)	PVBusSlave
ARMCortexA520CT.ext_bus	PVBusLogger
ARMCortexA520CT.ext_bus.mapper	PVBusMapper
ARMCortexA520CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

## Ports for ARMCortexA520CT

Port	Direction	Protocol	Description
acp_s	slave	PVBus	AXI ACP slave port.
AEND0MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND2MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND3MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).

Port	Direction	Protocol	Description
ASTART2MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	slave	Signal	CHI defined pins.
broadcastcachemaint	slave	Signal	ACE defined pins.
broadcastouter	slave	Signal	ACE defined pins.
broadcastpersist	slave	Signal	CHI defined pins.
cfgend	slave	Signal	This signal is for EE bit initialisation.
cfgsdisable	slave	Signal	This signal disables write access to some secure Interrupt Controller registers.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcs_m_pchannel	master	PChannel	Cluster PCSM signal
cluster_powerdown_out	master	Signal	DEPRECATED - An output from PPUs that informs thermal controller of the cluster power info
cluster_ppu_hw_stat	master	Value	Cluster PPU power state
clusterid	slave	Value	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set ManagerID64 of the core.
clusterpmuirq	master	Signal	DynamlQ pmu irq
CNTHPIRQ	master	Signal	Timer signals to SOC.
CNTHPSIRQ	master	Signal	Timer signals to SOC
CNTHVIRQ	master	Signal	Timer signals to SOC.
CNTHVSIRQ	master	Signal	Timer signals to SOC
CNTPNSIRQ	master	Signal	Timer signals to SOC.
CNTPSIRQ	master	Signal	Timer signals to SOC.
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.
CNTVIRQ	master	Signal	Timer signals to SOC.
commirq	master	Signal	Interrupt signal from debug communications channel.
core_clk_in	slave	ClockSignal	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
core_pcs_m_pchannel	master	PChannel	Core PCSM signals
core_powerdown_out	master	Signal	DEPRECATED - An output from PPUs that informs thermal controller of the core power info

Port	Direction	Protocol	Description
core_ppu_hw_stat	master	Value	Notify the power state change inside the PPU.
coreerrirq	master	Signal	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq	master	Signal	Fault indicator for a detected 1 or 2 bit ECC error
coreinstrrun	master	Signal	Core Running State
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
ctidbgirq	master	Signal	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	slave	Signal	External debug interface.
dbgnopwrdown	master	Signal	No power-down request.
dbgpwrupreq	master	Signal	Debug power up request.
defaultMP	slave	Signal	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main master interface ports.
dev_debug_s	slave	PVBus	External debug interface.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.
gicreset	master	Signal	An output from PPU that can be used to reset GICCLK domain components
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	slave	PVBus	External debug interface.
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	master	Signal	PPU cluster interrupt.
ppu_cluster_isolate	master	Signal	PPU Cluster Domain isolation control.
ppu_cluster_wakerequest	slave	Signal	PPU cluster wake request signal.
ppu_core_irq	master	Signal	PPU core interrupt.
ppu_core_isolate	master	Signal	PPU Core Domain isolation control.
ppu_core_wakerequest	slave	Signal	PPU core wake request signal.
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	master	PVBus	The core will generate bus requests on this port.
pvbus_periph_m	master	PVBus	The core can generate peripheral bus request on this port.
rei	slave	Signal	Per core RAM Error Interrupt.
reset	slave	Signal	Reset signal to cluster.
rvbaraddr	slave	Value_64	Reset vector base address.
sci_m	master	SystemCoherencyInterface	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei	slave	Signal	Per core System Error physical pins.

Port	Direction	Protocol	Description
spiden	slave	Signal	External debug interface.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq	master	Signal	Interrupt signal from the trace buffer unit.
utility_bus_s	slave	PVBus	Utility bus slave
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfiq	slave	Signal	Virtualised FIQ.
virq	slave	Signal	Virtualised IRQ.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Per core virtual System Error physical pins.

## Parameters for ARMCortexA520CT

### **cpuX.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **cpuX.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **cpuX.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **cpuX.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`



**cpuX.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**cpuX.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**cpuX.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**cpuX.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**cpuX.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**cpuX.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**cpuX.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**cpuX.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**cpuX.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**cpuX.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

### **`cpuX.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

### **`cpuX.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **`cpuX.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **`cpuX.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`cpuX.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`cpuX.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`cpuX.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`cpuX.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`cpuX.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`cpuX.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`AEND0_DEFAULT`**

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

Type: `uint64_t`

Default value: `0x0`

### **`AEND1_DEFAULT`**

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

Type: `uint64_t`

Default value: 0x0

#### **AEND2\_DEFAULT**

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

Type: uint64\_t

Default value: 0x0

#### **AEND3\_DEFAULT**

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

Type: uint64\_t

Default value: 0x0

#### **ASTART0\_DEFAULT**

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

Type: uint64\_t

Default value: 0x0

#### **ASTART1\_DEFAULT**

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

Type: uint64\_t

Default value: 0x0

#### **ASTART2\_DEFAULT**

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

Type: uint64\_t

Default value: 0x0

#### **ASTART3\_DEFAULT**

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

Type: uint64\_t

Default value: `0x0`

### **BROADCASTATOMIC**

Enable broadcasting of atomic operation. The `broadcastatomic` signal will override this value if used.

Type: `bool`

Default value: `true`

### **BROADCASTCACHEMAINT**

Enable broadcasting of cache maintenance operations to downstream caches. The `broadcastcachemaint` signal will override this value if used.

Type: `bool`

Default value: `false`

### **BROADCASTOUTER**

Enable broadcasting of Outer Shareable transactions. The `broadcastouter` signal will override this value if used.

Type: `bool`

Default value: `false`

### **BROADCASTPERSIST**

Enable broadcasting of cache clean to the point of persistence operations. The `broadcastpersist` signal will override this value if used.

Type: `bool`

Default value: `true`

### **CLUSTER\_ID**

The Cluster ID maps to the affinity levels of the `MPIDR_EL1` register and is evaluated based on the `MPIDR_EL1` layout. If `MPIDR_EL1` supports 16-bit cluster affinity levels, bits [15:8] map to `IDRAFF3`, while bits [7:0] map to `IDRAFF2`. If `MPIDR_EL1` supports 24-bit cluster affinity levels, the bits [23:16] map to `IDRAFF3`, bits [15:8] map to `IDRAFF2`, and bits [7:0] map to `IDRAFF1`. This configuration also updates all relevant component `DEVAFF` registers and is used to set the `ManagerID64` of the core.

Type: `uint32_t`

Default value: `0x0`

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. This value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required.

Type: `uint8_t`

Default value: 1

**CPUCFR**

Value of CPU Configuration Register.

Type: `uint64_t`

Default value: 0x0

**DBGROMADDR**

Initialization value of DBGDRAR register. Bits[55:12] of this register specify the ROM table physical address.

Type: `uint64_t`

Default value: 0x0

**DBGROMADDRV**

If true, set bits[1:0] of the CP15 DBGDRAR to indicate that the address is valid.

Type: `bool`

Default value: `false`

**GICDISABLE**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`

Default value: `true`

**NUM\_CORES**

Number of cores per cluster.

Type: `uint8_t`

Default value: 1



**bus\_type**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

Type: uint8\_t

Default value: 0

**core\_cache\_protection**

core\_cache\_protection can change ERROFR, ERROPFGF and ERROPFGCTL fields. Possible values are: -1:Not implemented (by default), 0:Disabled, 1:Enabled.

Type: int8\_t

Default value: 1

**core\_complex\_mapping**

Defines Complex descriptions for platforms that support several Cores per Complex like Cortex-A510. JSON format: {"complex0": { "cores" : [ 0, 1 ], "l2-cache" : {"exists":1, "size":16MB}}, ... , "complexN": { "cores" : [<core\_list>], "l2-cache" : {"exists":1, "size":16MB}}} where <core\_list> is the list of cores in the complexN. Effective only when the parameter value is not empty.

Type: string

Default value: "{ \"complex0\": { \"cores\": [0, 1], \"l2-cache\" : { \"exists\":1, \"size\": \"16MB\" } }, \"complex1\": { \"cores\": [2, 3], \"l2-cache\" : { \"exists\":1, \"size\": \"16MB\" } }, \"complex2\": { \"cores\": [4, 5], \"l2-cache\" : { \"exists\":1, \"size\": \"16MB\" } }, \"complex3\": { \"cores\": [6, 7], \"l2-cache\" : { \"exists\":1, \"size\": \"16MB\" } }, \"complex4\": { \"cores\": [8, 9], \"l2-cache\" : { \"exists\":1, \"size\": \"16MB\" } }, \"complex5\": { \"cores\": [10, 11], \"l2-cache\" : { \"exists\":1, \"size\": \"16MB\" } }, \"complex6\": { \"cores\": [12, 13], \"l2-cache\" : { \"exists\":1, \"size\": \"16MB\" } } }

**core\_power\_on\_by\_default**

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPUs which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

Type: bool

Default value: false

**cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: uint32\_t

Default value: 0x1

**cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-maintenance\_latency**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-miss\_latency**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-prefetch\_enabled**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

Type: `bool`

Default value: `false`

**dcache-read\_access\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-size**

L1 D-Cache size in bytes.

Type: `uint32_t`

Default value: `0x8000`

**dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-state\_modelled**

Set whether D-cache has stateful implementation.

Type: `bool`

Default value: `false`

**dcache-write\_access\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **enable\_simulation\_performance\_optimizations**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

Type: bool

Default value: true

### **ete.CLAIMTAGS**

Number of claim tags.

Type: uint8\_t

Default value: 4

### **ete.MAX\_INST\_PER\_Q**

Maximum limit for the number of instructions implied by a Q element.

Type: uint16\_t

Default value: 0x1

### **ete.PIDR\_CM0D**

TRCPIDR CM0D value.

Type: uint8\_t

Default value: 0

### **ete.PIDR\_REVAND**

TRCPIDR REVAND value.

Type: uint8\_t

Default value: 0

### **ete.PIDR\_REVISION**

TRCPIDR REVISION value.

Type: uint8\_t

Default value: 0

**ete.Q\_CADENCE**

Number of instruction blocks traced between two Q elements.

Type: `uint16_t`

Default value: `0x1`

**ete.RES0\_STATEFUL**

Whether **RES0** bits are stateful or **RAZ/WI**.

Type: `bool`

Default value: `false`

**ete.RETSTACK**

Return stack depth.

Type: `uint8_t`

Default value: `3`

**ete.REVISION**

TRCIDR1 revision value.

Type: `uint8_t`

Default value: `0`

**ete.SIM\_OVERFLOW\_GRANULARITY**

Number of instruction blocks in each granule, for simulated overflow.

Type: `uint32_t`

Default value: `0x64`

**ete.SIM\_OVERFLOW\_PERCENTAGE**

Percentage of instruction blocks lost in each granule, for simulated overflow.

Type: `uint8_t`

Default value: `0`

**ete.SOURCE\_ADDRESS**

Allow generation of source address elements.

Type: `bool`

Default value: `false`

**ete.TRACE\_OUTPUT**

File to which to write trace byte stream.

Type: `string`

Default value: `N/A`

**ete.TRCSRSTA\_FORCED\_EXCEP**

TRCSRSTA value for a forcibly traced exception.

Type: `bool`

Default value: `false`

**force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks**

Force MTE tag accesses to **RAZ/WI** and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

Type: `bool`

Default value: `false`

**force\_zero\_PSTATE\_PAN**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

**force\_zero\_mpam\_partid\_and\_pmg**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAMO\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

Type: `uint8_t`

Default value: `0`

**has\_actlr2**

If true ACTLR2 exists and ACTLR2(NS) is aliased to ACTLR\_EL1[63:32].

Type: `bool`

Default value: `true`

**has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

**has\_dot\_product**

Possible values of this parameter are:- 1, feature not implemented.- 2, feature is implemented.

Type: `uint8_t`

Default value: 2

**has\_ete**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (-plugin or -P).

Type: `bool`

Default value: `false`

**has\_peripheral\_port**

If true, an additional AXI peripheral port is configured.

Type: `bool`

Default value: `false`

**icache-hit\_latency**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`icache-miss_latency`**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`icache-prefetch_enabled`**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

### **`icache-read_access_latency`**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`icache-read_latency`**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`icache-size`**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: `0x8000`



**icache-state\_modelled**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

**invalidate\_code\_cache\_on\_icache\_cmo**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: `uint8_t`

Default value: `0`

**l3cache-has\_mpam**

L3 Cache has MPAM support.

Type: `bool`

Default value: `true`

**l3cache-hit\_latency**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-maintenance\_latency**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-miss\_latency**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-read\_access\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size l3cache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**l3cache-read\_bus\_width\_in\_bytes**

L3 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**l3cache-read\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. l3cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**l3cache-size**

L3 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**l3cache-snoop\_data\_transfer\_latency**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**l3cache-snoop\_issue\_latency**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **l3cache-ways**

L3 Cache number of ways (sets are implicit from size).

Type: `uint16_t`

Default value: 16

### **l3cache-write\_access\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size `l3cache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **l3cache-write\_bus\_width\_in\_bytes**

L3 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: `uint32_t`

Default value: 0x10

### **l3cache-write\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l3cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **memory\_tagging\_support\_level**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

Type: `uint8_t`

Default value: 3

### **mpmm\_accumulator\_multiplier**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator.value of n means the accumulator will use (n \* accumulator value) to calculate the mpmm threshold (MPMM).is provided as a fast model workaround to handle cases where execution of quantums

in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

Type: `uint8_t`

Default value: 1

### **num\_acp**

Number of ACP ports.

Type: `uint8_t`

Default value: 0

### **num\_nodes**

Number of transport nodes. Zero implies direct-connect configuration.

Type: `uint8_t`

Default value: 1

### **pmu-num\_counters**

Number of PMU counters implemented.

Type: `uint8_t`

Default value: 6

### **ptw\_latency**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

### **ras\_extra\_configurations**

Miscellaneous configurations for error records. An array of JSON objects. Note for `ERXCTLR_EL1` register it only allows to define the mask value for the `IMPDEF` fields, ie bits [63:32] and bit 1, but its reset value applies on all fields. Note for `ERXMISCN` masks - these are 64 bit masks covering the 64 bit registers `ERXMISCN_EL1`. E.g. [{"Index": 0, "ERXMISCO\_mask": 0x0, "ERXMISCO\_reset": 0x0, "ERXMISC1\_mask": 0x0, "ERXMISC1\_reset": 0x0, "ERXMISC2\_mask": 0x0, "ERXMISC2\_reset": 0x0, "ERXMISC3\_mask": 0x0, "ERXMISC3\_reset": 0x0, "ERXCTLR\_EL1\_mask": 0x0, "ERXCTLR\_EL1\_reset": 0x0}, {"Index": 1, "ERXMISCO\_mask": 0x0, "ERXMISCO\_reset": 0x0, "ERXSTATUS\_IERR\_mask": 0x300}].

Type: `string`

Default value: "[ { \"Index\": 1, \"ERXMISC0\_mask\": 0xFFFFc0003fc3, \"ERXMISC1\_mask\": 0x03F870003FF30f07, \"ERXPGCTL\_reset\": 0x1000 }, { \"Index\": 2, \"ERXMISC0\_mask\": 0xFFFFe007ffc0, \"ERXMISC0\_reset\": 0x2, \"ERXSTATUS\_IERR\_mask\": 0x300 , \"ERXMISC1\_mask\": 0x0FF8700ffff31f0f, \"ERXPGCTL\_reset\": 0x1000 } ]"

### **ras\_pfg\_clock\_mhz**

RAS Pseudo-Fault generation clock rate in MHz.

Type: `uint8_t`

Default value: 12

### **revision\_number**

Revision number of TRM implemented by the model. Changing the value can change the model behaviour. Visible as the Variant field in MIDR/MIDR\_EL1. Corresponds to the revision number X in rXpY.

Type: `uint8_t`

Default value: 0

### **tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

### **tlbi\_stall\_enabled**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: `bool`

Default value: `false`

### **treat\_PAC\_as\_NOP**

Non-architecture parameter. Treat Pointer Authentication as **NOP**. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are **NOP**. Branch, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

Type: `bool`

Default value: `false`

### **walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

## 3.51 ARMCortexA520CT\_CortexA720CT

Defined in LISA/ARMCortexA520CT\_CortexA720CT.lisa.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
CortexA520 r0p1	Full support
CortexA720 r0p1	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following parameters were added:

- num\_acp
- subcluster0.has\_delayed\_dbgreg
- subcluster0.has\_delayed\_sysreg
- subcluster1.has\_delayed\_dbgreg
- subcluster1.has\_delayed\_sysreg

### About ARMCortexA520CT\_CortexA720CT

The number of cores in each subcluster is configurable using the following parameters:

#### **subcluster0.NUM\_CORES**

Possible values are 1-13 (ARMCortexA520CT).

#### **subcluster1.NUM\_CORES**

Possible values are 1-13 (ARMCortexA720CT).

The total number of cores in the cluster cannot exceed 14.

Port arrays are expanded to the width needed for the maximum number of cores in each subcluster. Use the following port array indexes:

- <port\_name>[0-12] for cores in subcluster0.
- <port\_name>[13-25] for cores in subcluster1.



All instances in the Master cross trigger matrix port array, `cti[26]` must be connected, regardless of the `NUM_CORES` value used.

Core-specific parameters have the following prefixes:

- `subcluster0.cpu0` to `subcluster0.cpu12` identify cores in `subcluster0`.
- `subcluster1.cpu0` to `subcluster1.cpu12` identify cores in `subcluster1`.

For information about the cores in this model, see:

- [ARMCortexA520CT](#).
- [ARMCortexA720CT](#).

### Iris and MTI instances for ARMCortexA520CT\_CortexA720CT

This model has the following Iris instances:

Name	Instance type
ARMCortexA520CT_CortexA720CT	Cluster_ARM_CortexA520_CortexA720_Heterogeneous
ARMCortexA520CT_CortexA720CT.AMU	PVBusLogger
ARMCortexA520CT_CortexA720CT.AMU.mapper	PVBusMapper
ARMCortexA520CT_CortexA720CT.DAP	PVBusLogger
ARMCortexA520CT_CortexA720CT.DAP.mapper	PVBusMapper
ARMCortexA520CT_CortexA720CT.DSU	DSU-120
ARMCortexA520CT_CortexA720CT.DSU.PPU_cluster	PPUv1
ARMCortexA520CT_CortexA720CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA520CT_CortexA720CT.DSU.PPU_coreZ (where Z = 0-1)	PPUv1
ARMCortexA520CT_CortexA720CT.DSU.PPU_coreZ.busslave (where Z = 0-1)	PVBusSlave
ARMCortexA520CT_CortexA720CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA520CT_CortexA720CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA520CT_CortexA720CT.DSU.shared_cache	PVCache
ARMCortexA520CT_CortexA720CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA520CT_CortexA720CT.DSU.shared_cache.upstream[Z] (where Z = 0-6)	PVBusSlave
ARMCortexA520CT_CortexA720CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA520CT_CortexA720CT.MMAP	PVBusLogger
ARMCortexA520CT_CortexA720CT.MMAP.mapper	PVBusMapper
ARMCortexA520CT_CortexA720CT.RAS	PVBusLogger
ARMCortexA520CT_CortexA720CT.RAS.mapper	PVBusMapper
ARMCortexA520CT_CortexA720CT.cpuZ.debug_rom (where Z = 0-1)	debug_rom
ARMCortexA520CT_CortexA720CT.ext_bus	PVBusLogger

Name	Instance type
ARMCortexA520CT_CortexA720CT.ext_bus.mapper	PVBusMapper
ARMCortexA520CT_CortexA720CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA520CT_CortexA720CT.global_debug_rom	debug_rom
ARMCortexA520CT_CortexA720CT.secondary_debug_rom	debug_rom
ARMCortexA520CT_CortexA720CT.subcluster0	Subcluster_ARM_Cortex-A520
ARMCortexA520CT_CortexA720CT.subcluster0.cpu0	ARM_Cortex-A520
ARMCortexA520CT_CortexA720CT.subclusterZ.cpu0.UTLB (where Z = 0-1)	TLB
ARMCortexA520CT_CortexA720CT.subclusterZ.cpuU.dtlb (where Z = 0-1; U = 0-1)	TLB
ARMCortexA520CT_CortexA720CT.subclusterZ.cpu0.l1dcache (where Z = 0-1)	PVCache
ARMCortexA520CT_CortexA720CT.subclusterZ.cpu0.l1dcache.downstream[0].pvbusmaster (where Z = 0-1)	PVBusMaster
ARMCortexA520CT_CortexA720CT.subclusterZ.cpu0.l1dcache.upstream[0] (where Z = 0-1)	PVBusSlave
ARMCortexA520CT_CortexA720CT.subclusterZ.cpu0.l1icache (where Z = 0-1)	PVCache
ARMCortexA520CT_CortexA720CT.subclusterZ.cpu0.l1icache.downstream[0].pvbusmaster (where Z = 0-1)	PVBusMaster
ARMCortexA520CT_CortexA720CT.subclusterZ.cpu0.l1icache.upstream[0] (where Z = 0-1)	PVBusSlave
ARMCortexA520CT_CortexA720CT.subclusterZ.cpu0.l2cache (where Z = 0-1)	PVCache
ARMCortexA520CT_CortexA720CT.subclusterZ.cpu0.l2cache.downstream[0].pvbusmaster (where Z = 0-1)	PVBusMaster
ARMCortexA520CT_CortexA720CT.subclusterZ.cpu0.l2cache.upstream[W] (where Z = 0-1; W = 0-1)	PVBusSlave
ARMCortexA520CT_CortexA720CT.subclusterZ.sve (where Z = 0-1)	ScalableVectorExtension
ARMCortexA520CT_CortexA720CT.subcluster1	Subcluster_ARM_Cortex-A720
ARMCortexA520CT_CortexA720CT.subcluster1.cpu0	ARM_Cortex-A720

This model has the following MTI trace components:

Name	Component type
ARMCortexA520CT_CortexA720CT.AMU	PVBusLogger
ARMCortexA520CT_CortexA720CT.AMU.mapper	PVBusMapper
ARMCortexA520CT_CortexA720CT.DAP	PVBusLogger
ARMCortexA520CT_CortexA720CT.DAP.mapper	PVBusMapper
ARMCortexA520CT_CortexA720CT.DSU	DSU-120
ARMCortexA520CT_CortexA720CT.DSU.PPU_cluster	PPUv1
ARMCortexA520CT_CortexA720CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA520CT_CortexA720CT.DSU.PPU_coreZ (where Z = 0-1)	PPUv1
ARMCortexA520CT_CortexA720CT.DSU.PPU_coreZ.busslave (where Z = 0-1)	PVBusSlave



Name	Component type
ARMCortexA520CT_CortexA720CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA520CT_CortexA720CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA520CT_CortexA720CT.DSU.shared_cache	PVCache
ARMCortexA520CT_CortexA720CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA520CT_CortexA720CT.DSU.shared_cache.upstream[Z] (where Z = 0-6)	PVBusSlave
ARMCortexA520CT_CortexA720CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA520CT_CortexA720CT.MMAP	PVBusLogger
ARMCortexA520CT_CortexA720CT.MMAP.mapper	PVBusMapper
ARMCortexA520CT_CortexA720CT.RAS	PVBusLogger
ARMCortexA520CT_CortexA720CT.RAS.mapper	PVBusMapper
ARMCortexA520CT_CortexA720CT.ext_bus	PVBusLogger
ARMCortexA520CT_CortexA720CT.ext_bus.mapper	PVBusMapper
ARMCortexA520CT_CortexA720CT.gic_cpuif_decoder_cluster	GlCv3CPUInterfaceDecoder
ARMCortexA520CT_CortexA720CT.subcluster0.cpu0	ARM_Cortex-A520
ARMCortexA520CT_CortexA720CT.subclusterZ.cpu0.UTLB (where Z = 0-1)	TLB
ARMCortexA520CT_CortexA720CT.subclusterZ.cpu0.l1dcache (where Z = 0-1)	PVCache
ARMCortexA520CT_CortexA720CT.subclusterZ.cpu0.l1dcache.downstream[0].pvbusmaster (where Z = 0-1)	PVBusMaster
ARMCortexA520CT_CortexA720CT.subclusterZ.cpu0.l1dcache.upstream[0] (where Z = 0-1)	PVBusSlave
ARMCortexA520CT_CortexA720CT.subclusterZ.cpu0.l1icache (where Z = 0-1)	PVCache
ARMCortexA520CT_CortexA720CT.subclusterZ.cpu0.l1icache.downstream[0].pvbusmaster (where Z = 0-1)	PVBusMaster
ARMCortexA520CT_CortexA720CT.subclusterZ.cpu0.l1icache.upstream[0] (where Z = 0-1)	PVBusSlave
ARMCortexA520CT_CortexA720CT.subclusterZ.cpu0.l2cache (where Z = 0-1)	PVCache
ARMCortexA520CT_CortexA720CT.subclusterZ.cpu0.l2cache.downstream[0].pvbusmaster (where Z = 0-1)	PVBusMaster
ARMCortexA520CT_CortexA720CT.subclusterZ.cpu0.l2cache.upstream[W] (where Z = 0-1; W = 0-1)	PVBusSlave
ARMCortexA520CT_CortexA720CT.subcluster1.cpu0	ARM_Cortex-A720

### Ports for ARMCortexA520CT\_CortexA720CT

Port	Direction	Protocol	Description
acp_s	slave	PVBus	AXI ACP slave port.
AEND0MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND2MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND3MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).

Port	Direction	Protocol	Description
ASTART0MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART2MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	slave	Signal	CHI defined pins.
broadcastcachemaint	slave	Signal	ACE defined pins.
broadcastouter	slave	Signal	ACE defined pins.
broadcastpersist	slave	Signal	CHI defined pins.
cfgend	slave	Signal	This signal is for EE bit initialisation.
cfgsdisable	slave	Signal	This signal disables write access to some secure Interrupt Controller registers.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcsn_pchannel	master	PChannel	Cluster PCSM signal
cluster_powerdown_out	master	Signal	DEPRECATED - An output from PPU that informs thermal controller of the cluster power info
cluster_ppu_hw_stat	master	Value	Cluster PPU power state
clusterid	slave	Value	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set ManagerID64 of the core.
clusterpmuirq	master	Signal	DynamiQ pmu irq
CNTHPIRQ	master	Signal	Timer signals to SOC.
CNTHPSIRQ	master	Signal	Timer signals to SOC
CNTHVIRQ	master	Signal	Timer signals to SOC.
CNTHVSIRQ	master	Signal	Timer signals to SOC
CNTPNSIRQ	master	Signal	Timer signals to SOC.
CNTPSIRQ	master	Signal	Timer signals to SOC.
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.
CNTVIRQ	master	Signal	Timer signals to SOC.
commirq	master	Signal	Interrupt signal from debug communications channel.

Port	Direction	Protocol	Description
core_clk_in	slave	ClockSignal	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
core_pcsn_pchannel	master	PChannel	Core PCSM signals
core_powerdown_out	master	Signal	DEPRECATED - An output from PPU that informs thermal controller of the core power info
core_ppu_hw_stat	master	Value	Notify the power state change inside the PPU.
coreerrirq	master	Signal	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq	master	Signal	Fault indicator for a detected 1 or 2 bit ECC error
coreinstrrun	master	Signal	Core Running State
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
ctidbgirq	master	Signal	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	slave	Signal	External debug interface.
dbgnopwrdown	master	Signal	No power-down request.
dbgpwrupreq	master	Signal	Debug power up request.
defaultMP	slave	Signal	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main master interface ports.
dev_debug_s	slave	PVBus	External debug interface.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.
gicreset	master	Signal	An output from PPU that can be used to reset GICCLK domain components
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	slave	PVBus	External debug interface.
pmbirq	master	Signal	Interrupt signal from the statistical profiling unit.
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	master	Signal	PPU cluster interrupt.
ppu_cluster_isolate	master	Signal	PPU Cluster Domain isolation control.
ppu_cluster_wakerequest	slave	Signal	PPU cluster wake request signal.
ppu_core_irq	master	Signal	PPU core interrupt.
ppu_core_isolate	master	Signal	PPU Core Domain isolation control.
ppu_core_wakerequest	slave	Signal	PPU core wake request signal.
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	master	PVBus	The core will generate bus requests on this port.

Port	Direction	Protocol	Description
pvbus_periph_m	master	PVBus	The core can generate peripheral bus request on this port.
rei	slave	Signal	Per core RAM Error Interrupt.
reset	slave	Signal	Reset signal to cluster.
rvbaraddr	slave	Value_64	Reset vector base address.
sci_m	master	SystemCoherencyInterface	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei	slave	Signal	Per core System Error physical pins.
spiden	slave	Signal	External debug interface.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq	master	Signal	Interrupt signal from the trace buffer unit.
utility_bus_s	slave	PVBus	Utility bus slave
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfiq	slave	Signal	Virtualised FIQ.
virq	slave	Signal	Virtualised IRQ.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Per core virtual System Error physical pins.

## Parameters for ARM CortexA520CT\_CortexA720CT

### AEND0\_DEFAULT

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

Type: uint64\_t

Default value: 0x0

### AEND1\_DEFAULT

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

Type: uint64\_t

Default value: 0x0

### AEND2\_DEFAULT

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

Type: uint64\_t

Default value: 0x0

### **AEND3\_DEFAULT**

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

Type: uint64\_t

Default value: 0x0

### **ASTART0\_DEFAULT**

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

Type: uint64\_t

Default value: 0x0

### **ASTART1\_DEFAULT**

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

Type: uint64\_t

Default value: 0x0

### **ASTART2\_DEFAULT**

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

Type: uint64\_t

Default value: 0x0

### **ASTART3\_DEFAULT**

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

Type: uint64\_t

Default value: 0x0

### **BROADCASTATOMIC**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: bool

Default value: `true`

### **BROADCASTCACHEMAINT**

Enable broadcasting of cache maintenance operations to downstream caches. The `broadcastcachemaint` signal will override this value if used.

Type: `bool`

Default value: `false`

### **BROADCASTOUTER**

Enable broadcasting of Outer Shareable transactions. The `broadcastouter` signal will override this value if used.

Type: `bool`

Default value: `false`

### **BROADCASTPERSIST**

Enable broadcasting of cache clean to the point of persistence operations. The `broadcastpersist` signal will override this value if used.

Type: `bool`

Default value: `true`

### **CLUSTER\_ID**

The Cluster ID maps to the affinity levels of the `MPIDR_EL1` register and is evaluated based on the `MPIDR_EL1` layout. If `MPIDR_EL1` supports 16-bit cluster affinity levels, bits [15:8] map to `IDRAFF3`, while bits [7:0] map to `IDRAFF2`. If `MPIDR_EL1` supports 24-bit cluster affinity levels, the bits [23:16] map to `IDRAFF3`, bits [15:8] map to `IDRAFF2`, and bits [7:0] map to `IDRAFF1`. This configuration also updates all relevant component `DEVAFF` registers and is used to set the `ManagerID64` of the core.

Type: `uint32_t`

Default value: `0x0`

### **CMO\_broadcast\_when\_cache\_state\_modelling\_disabled**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. This value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain events even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required.

Type: `uint8_t`

Default value: 1

### **DBGROMADDR**

Initialization value of DBGDRAR register. Bits[55:12] of this register specify the ROM table physical address.

Type: `uint64_t`

Default value: `0x0`

### **DBGROMADDRV**

If true, set bits[1:0] of the CP15 DBGDRAR to indicate that the address is valid.

Type: `bool`

Default value: `false`

### **GICDISABLE**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`

Default value: `true`

### **bus\_type**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

Type: `uint8_t`

Default value: 0

### **core\_complex\_mapping**

Defines Complex descriptions for platforms that support several Cores per Complex like Cortex-A510. JSON format: {"complex0": { "cores" : [ 0, 1 ], "l2-cache" : {"exists":1, "size":16MB}}, ... , "complexN": { "cores" : [<core\_list>, "l2-cache" : {"exists":1, "size":16MB}}} where <core\_list> is the list of cores in the complexN. Effective only when the parameter value is not empty.

Type: `string`

Default value: "{ \"complex0\": { \"cores\": [0, 1], \"l2-cache\" : { \"exists\":1, \"size\": \"16MB\" } }, \"complex1\": { \"cores\": [2, 3], \"l2-cache\" : { \"exists\":1, \"size\": \"16MB\" } }, \"complex2\": { \"cores\": [4, 5], \"l2-cache\" : { \"exists\":1, \"size\": \"16MB\" } }, \"complex3\": { \"cores\": [6, 7], \"l2-cache\" : { \"exists\":1, \"size\": \"16MB\" } }, \"complex4\": { \"cores\": [8, 9], \"l2-cache\" : { \"exists\":1, \"size\": \"16MB\" } }, \"complex5\": { \"cores\": [10, 11], \"l2-cache\" : { \"exists\":1, \"size\": \"16MB\" } }, \"complex6\": { \"cores\": [12, 13], \"l2-cache\" : { \"exists\":1, \"size\": \"16MB\" } } }

**core\_power\_on\_by\_default**

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPUs which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

Type: `bool`

Default value: `false`

**dcache-state\_modelled**

Set whether D-cache has stateful implementation.

Type: `bool`

Default value: `false`

**enable\_simulation\_performance\_optimizations**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of `stage12_tlb_size` parameter to 1024).

Type: `bool`

Default value: `true`

**has\_peripheral\_port**

If true, an additional AXI peripheral port is configured.

Type: `bool`

Default value: `false`

**icache-state\_modelled**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

**l3cache-has\_mpam**

L3 Cache has MPAM support.

Type: `bool`

Default value: `false`



**l3cache-hit\_latency**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-maintenance\_latency**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-miss\_latency**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-read\_access\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size `l3cache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-read\_bus\_width\_in\_bytes**

L3 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: `uint32_t`

Default value: `0x10`

**l3cache-read\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l3cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **l3cache-size**

L3 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

### **l3cache-snoop\_data\_transfer\_latency**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **l3cache-snoop\_issue\_latency**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **l3cache-ways**

L3 Cache number of ways (sets are implicit from size).

Type: `uint16_t`

Default value: `16`

### **l3cache-write\_access\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size `l3cache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **l3cache-write\_bus\_width\_in\_bytes**

L3 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: `uint32_t`

Default value: 0x10

### **l3cache-write\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. l3cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **mpmm\_accumulator\_multiplier**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator.value of n means the accumulator will use (n \* accumulator value) to calculate the mpmm threshold (MPMM).is provided as a fast model workaround to handle cases where execution of quantum in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

Type: uint8\_t

Default value: 1

### **num\_acp**

Number of ACP ports.

Type: uint8\_t

Default value: 0

### **num\_nodes**

Number of transport nodes. Zero implies direct-connect configuration.

Type: uint8\_t

Default value: 1

### **revision\_number**

Revision number of TRM implemented by the model. Changing the value can change the model behaviour. Visible as the Variant field in MIDR/MIDR\_EL1. Corresponds to the revision number X in rXpY.

Type: uint8\_t

Default value: 0

### **subcluster0.CPUCFR**

Value of CPU Configuration Register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.NUM_CORES`**

Number of cores in the subcluster. Total number of cores in cluster may not exceed 14.

Type: `uint8_t`

Default value: `1`

### **`subcluster0.core_cache_protection`**

`core_cache_protection` can change `ERROFR`, `ERROPFGF` and `ERROPFGCTL` fields. Possible values are: -1:Not implemented (by default), 0:Disabled, 1:Enabled.

Type: `int8_t`

Default value: `1`

### **`subcluster0.cpi_div`**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

### **`subcluster0.cpi_mul`**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

### **`subcluster0.cpu0.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu0.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster0.cpu0.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu0.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu0.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu0.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu0.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu0.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu0.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu0.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu0.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu0.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu0.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu0.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu0.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu0.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu0.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu0.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster0.cpu0.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster0.cpu0.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu0.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu0.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`



**subcluster0.cpu0.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu0.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu0.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu0.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu0.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu0.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster0.cpu0.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu1.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu1.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu1.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu1.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu1.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu1.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu1.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu1.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu1.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu1.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu1.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu1.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu1.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu1.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu1.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu1.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster0.cpu1.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster0.cpu1.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster0.cpu1.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster0.cpu1.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu1.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu1.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu1.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu1.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu1.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster0.cpu1.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu1.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster0.cpu10.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu10.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu10.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu10.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu10.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu10.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu10.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu10.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu10.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu10.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t



Default value: 0x0

### **subcluster0.cpu10.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

### **subcluster0.cpu10.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster0.cpu10.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster0.cpu10.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

### **subcluster0.cpu10.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster0.cpu10.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu10.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu10.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu10.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu10.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu10.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu10.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu10.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

### **`subcluster0.cpu10.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

### **`subcluster0.cpu10.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **`subcluster0.cpu10.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **`subcluster0.cpu10.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster0.cpu10.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu10.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu10.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster0.cpu10.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu10.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu10.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu10.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu11.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu11.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster0.cpu11.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu11.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu11.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu11.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu11.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu11.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster0.cpu11.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu11.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu11.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu11.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu11.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu11.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu11.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu11.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu11.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu11.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu11.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu11.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu11.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu11.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu11.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster0.cpu11.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster0.cpu11.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu11.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu11.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`



**subcluster0.cpu11.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu11.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu11.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster0.cpu11.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu11.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu11.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster0.cpu11.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster0.cpu12.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu12.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu12.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu12.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu12.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu12.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu12.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu12.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu12.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu12.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu12.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu12.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu12.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu12.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu12.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu12.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu12.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu12.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu12.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu12.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu12.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu12.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu12.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster0.cpu12.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster0.cpu12.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster0.cpu12.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster0.cpu12.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu12.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu12.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu12.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu12.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu12.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu12.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu12.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu2.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu2.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu2.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu2.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu2.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu2.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t



Default value: 0x0

#### **subcluster0.cpu2.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

#### **subcluster0.cpu2.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu2.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu2.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

#### **subcluster0.cpu2.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu2.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu2.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu2.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu2.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu2.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu2.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu2.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**`subcluster0.cpu2.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**`subcluster0.cpu2.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**`subcluster0.cpu2.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**`subcluster0.cpu2.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**`subcluster0.cpu2.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**`subcluster0.cpu2.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**`subcluster0.cpu2.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster0.cpu2.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu2.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu2.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu2.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu3.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu3.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu3.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu3.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu3.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu3.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu3.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu3.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster0.cpu3.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu3.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu3.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu3.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu3.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu3.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu3.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu3.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu3.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu3.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster0.cpu3.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster0.cpu3.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu3.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu3.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`



**subcluster0.cpu3.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu3.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster0.cpu3.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu3.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu3.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster0.cpu3.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster0.cpu4.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu4.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu4.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu4.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu4.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu4.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu4.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu4.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu4.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu4.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu4.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu4.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu4.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu4.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu4.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu4.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster0.cpu4.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster0.cpu4.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster0.cpu4.semihosting-cwd**

Base directory for semihosting file access.

Type: string

Default value: N/A

**subcluster0.cpu4.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: true

**subcluster0.cpu4.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu4.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster0.cpu4.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu4.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster0.cpu4.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu4.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster0.cpu5.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu5.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu5.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu5.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu5.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu5.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t



Default value: 0x0

**subcluster0.cpu5.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu5.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu5.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu5.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu5.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu5.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu5.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu5.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu5.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**`subcluster0.cpu5.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**`subcluster0.cpu5.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**`subcluster0.cpu5.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**`subcluster0.cpu5.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**`subcluster0.cpu5.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**`subcluster0.cpu5.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**`subcluster0.cpu5.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster0.cpu5.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu5.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu5.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu5.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu6.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu6.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu6.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu6.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu6.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu6.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu6.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu6.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster0.cpu6.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu6.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu6.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu6.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu6.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu6.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu6.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu6.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu6.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu6.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster0.cpu6.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster0.cpu6.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu6.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu6.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`



**subcluster0.cpu6.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu6.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster0.cpu6.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu6.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu6.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster0.cpu6.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu7.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu7.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu7.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu7.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu7.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu7.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu7.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu7.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu7.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu7.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu7.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu7.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu7.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu7.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu7.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu7.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster0.cpu7.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster0.cpu7.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster0.cpu7.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster0.cpu7.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu7.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu7.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu7.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu7.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu7.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster0.cpu7.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu7.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster0.cpu8.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu8.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu8.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu8.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu8.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu8.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t



Default value: 0x0

**subcluster0.cpu8.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu8.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu8.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu8.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu8.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu8.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu8.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu8.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu8.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**`subcluster0.cpu8.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**`subcluster0.cpu8.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**`subcluster0.cpu8.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**`subcluster0.cpu8.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**`subcluster0.cpu8.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**`subcluster0.cpu8.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**`subcluster0.cpu8.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster0.cpu8.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu8.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu8.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu8.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu9.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu9.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu9.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu9.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu9.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu9.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu9.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu9.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster0.cpu9.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu9.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu9.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu9.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu9.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu9.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu9.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu9.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu9.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu9.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster0.cpu9.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster0.cpu9.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu9.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu9.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`



**subcluster0.cpu9.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu9.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster0.cpu9.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu9.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu9.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster0.cpu9.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.dcache-hit_latency`**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.dcache-maintenance_latency`**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.dcache-miss_latency`**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.dcache-prefetch_enabled`**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

Type: `bool`

Default value: `false`

### **`subcluster0.dcache-read_access_latency`**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.dcache-size**

L1 D-Cache size in bytes.

Type: `uint32_t`

Default value: `0x8000`

**subcluster0.dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.dcache-write\_access\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.etc.CLAIMTAGS**

Number of claim tags.

Type: `uint8_t`

Default value: 4

**subcluster0.ete.MAX\_INST\_PER\_Q**

Maximum limit for the number of instructions implied by a Q element.

Type: `uint16_t`

Default value: `0x1`

**subcluster0.ete.PIDR\_CM0D**

TRCPIDR CM0D value.

Type: `uint8_t`

Default value: 0

**subcluster0.ete.PIDR\_REVAND**

TRCPIDR REVAND value.

Type: `uint8_t`

Default value: 0

**subcluster0.ete.PIDR\_REVISION**

TRCPIDR REVISION value.

Type: `uint8_t`

Default value: 0

**subcluster0.ete.Q\_CADENCE**

Number of instruction blocks traced between two Q elements.

Type: `uint16_t`

Default value: `0x1`

**subcluster0.ete.RES0\_STATEFUL**

Whether **RES0** bits are stateful or **RAZ/WI**.

Type: `bool`

Default value: `false`

**subcluster0.ete.RETSTACK**

Return stack depth.

Type: `uint8_t`

Default value: 3

**subcluster0.ete.REVISION**

TRCIDR1 revision value.

Type: `uint8_t`

Default value: 0

**subcluster0.ete.SIM\_OVERFLOW\_GRANULARITY**

Number of instruction blocks in each granule, for simulated overflow.

Type: `uint32_t`

Default value: 0x64

**subcluster0.ete.SIM\_OVERFLOW\_PERCENTAGE**

Percentage of instruction blocks lost in each granule, for simulated overflow.

Type: `uint8_t`

Default value: 0

**subcluster0.ete.SOURCE\_ADDRESS**

Allow generation of source address elements.

Type: `bool`

Default value: `false`

**subcluster0.ete.TRACE\_OUTPUT**

File to which to write trace byte stream.

Type: `string`

Default value: N/A

**subcluster0.ete.TRCSRSTA\_FORCED\_EXCEP**

TRCSRSTA value for a forcibly traced exception.

Type: `bool`

Default value: `false`

**subcluster0.force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks**

Force MTE tag accesses to **RAZ/WI** and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/

GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

Type: `bool`

Default value: `false`

#### **`subcluster0.force_zero_PSTATE_PAN`**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

#### **`subcluster0.force_zero_mpam_partid_and_pmg`**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

Type: `uint8_t`

Default value: `0`

#### **`subcluster0.has_actlr2`**

If true ACLTR2 exists and ACTLR2(NS) is aliased to ACTLR\_EL1[63:32].

Type: `bool`

Default value: `true`

#### **`subcluster0.has_delayed_dbgreg`**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

#### **`subcluster0.has_delayed_sysreg`**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

#### **`subcluster0.has_dot_product`**

Possible values of this parameter are:- 1, feature not implemented.- 2, feature is implemented.

Type: `uint8_t`

Default value: 2

### **`subcluster0.has_ete`**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (`-plugin` or `-P`).

Type: `bool`

Default value: `false`

### **`subcluster0.icache-hit_latency`**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.icache-maintenance_latency`**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.icache-miss_latency`**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.icache-prefetch_enabled`**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

### **`subcluster0.icache-read_access_latency`**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and

intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.icache-read_latency`**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.icache-size`**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: `0x8000`

#### **`subcluster0.invalidate_code_cache_on_icache_cmo`**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: `uint8_t`

Default value: `0`

#### **`subcluster0.memory_tagging_support_level`**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

Type: `uint8_t`

Default value: `3`

#### **`subcluster0.pmu-num_counters`**

Number of PMU counters implemented.

Type: `uint8_t`

Default value: `6`



**subcluster0.ptw\_latency**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

**subcluster0.ras\_extra\_configurations**

Miscellaneous configurations for error records. An array of JSON objects. Note for ERXCTLR\_EL1 register it only allows to define the mask value for the IMPDEF fields, ie bits [63:32] and bit 1, but its reset value applies on all fields. Note for ERXMISCn masks - these are 64 bit masks covering the 64 bit registers ERXMISCn\_EL1. E.g. [{"Index": 0, "ERXMISCO\_mask": 0x0, "ERXMISCO\_reset": 0x0, "ERXMISC1\_mask": 0x0, "ERXMISC1\_reset": 0x0, "ERXMISC2\_mask": 0x0, "ERXMISC2\_reset": 0x0, "ERXMISC3\_mask": 0x0, "ERXMISC3\_reset": 0x0, "ERXCTLR\_EL1\_mask": 0x0, "ERXCTLR\_EL1\_reset": 0x0}, {"Index": 1, "ERXMISCO\_mask": 0x0, "ERXMISCO\_reset": 0x0, "ERXSTATUS\_IERR\_mask": 0x300}].

Type: string

Default value: "[ { \"Index\": 1, \"ERXMISCO\_mask\": 0xFFFFc0003fc3, \"ERXMISC1\_mask\": 0x03F870003FF30f07, \"ERXPGCTL\_reset\": 0x1000 }, { \"Index\": 2, \"ERXMISCO\_mask\": 0xFFFFe007ffc0, \"ERXMISCO\_reset\": 0x2, \"ERXSTATUS\_IERR\_mask\": 0x300 , \"ERXMISC1\_mask\": 0x0FF8700ffff31f0f, \"ERXPGCTL\_reset\": 0x1000 } ]"

**subcluster0.ras\_pfg\_clock\_mhz**

RAS Pseudo-Fault generation clock rate in MHz.

Type: uint8\_t

Default value: 12

**subcluster0.revision\_number**

Revision number of TRM implemented by the model. Changing the value can change the model behaviour. Visible as the Variant field in MIDR/MIDR\_EL1. Corresponds to the revision number X in rXpY.

Type: uint8\_t

Default value: 0

**subcluster0.tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

**subcluster0.tlbi\_stall\_enabled**

If true, tlbi invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: `bool`

Default value: `false`

**subcluster0.treat\_PAC\_as\_NOP**

Non-architecture parameter. Treat Pointer Authentication as **NOP**. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are **NOP**. Branch, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

Type: `bool`

Default value: `false`

**subcluster0.walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.CPUCFR**

Value of CPU Configuration Register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.NUM\_CORES**

Number of cores in the subcluster. Total number of cores in cluster may not exceed 14.

Type: `uint8_t`

Default value: `1`

**subcluster1.cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**subcluster1.cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**subcluster1.cpu0.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu0.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster1.cpu0.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu0.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu0.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu0.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

#### **`subcluster1.cpu0.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu0.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu0.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu0.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu0.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu0.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

#### **`subcluster1.cpu0.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu0.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu0.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu0.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu0.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

#### **`subcluster1.cpu0.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: 0

#### **`subcluster1.cpu0.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

#### **`subcluster1.cpu0.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

#### **`subcluster1.cpu0.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: 0x123456

#### **`subcluster1.cpu0.semihosting-T32_HLT`**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

#### **`subcluster1.cpu0.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

#### **`subcluster1.cpu0.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

#### **`subcluster1.cpu0.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

### **`subcluster1.cpu0.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster1.cpu0.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu0.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster1.cpu0.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster1.cpu0.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster1.cpu0.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu0.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu0.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster1.cpu1.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu1.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster1.cpu1.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu1.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu1.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.



Type: `bool`

Default value: `false`

### **`subcluster1.cpu1.force-fpsid`**

Override the FPSID value.

Type: `bool`

Default value: `true`

### **`subcluster1.cpu1.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to

the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu1.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

#### **`subcluster1.cpu1.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu1.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu1.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu1.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu1.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu1.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu1.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu1.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu1.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu1.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster1.cpu1.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster1.cpu1.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu1.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu1.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu1.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu1.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu1.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu1.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: 0xf000000

**subcluster1.cpu1.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu1.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster1.cpu1.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: true

**subcluster1.cpu10.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: false

**subcluster1.cpu10.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: bool

Default value: false

**subcluster1.cpu10.CRYPTODISABLE**

Disable cryptographic features.

Type: bool

Default value: false

**subcluster1.cpu10.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu10.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu10.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu10.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu10.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu10.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu10.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu10.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu10.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster1.cpu10.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu10.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu10.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu10.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu10.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

### **`subcluster1.cpu10.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

### **`subcluster1.cpu10.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu10.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu10.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`



**subcluster1.cpu10.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster1.cpu10.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster1.cpu10.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster1.cpu10.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster1.cpu10.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu10.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu10.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu10.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu10.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu10.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu10.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu10.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster1.cpu11.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu11.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster1.cpu11.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu11.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu11.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu11.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu11.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu11.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu11.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu11.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu11.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu11.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

### **`subcluster1.cpu11.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu11.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu11.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu11.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu11.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu11.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu11.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu11.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu11.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu11.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster1.cpu11.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster1.cpu11.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu11.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu11.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu11.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu11.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu11.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster1.cpu11.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu11.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu11.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster1.cpu11.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster1.cpu12.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu12.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu12.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu12.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu12.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu12.force-fpsid`**

Override the FPSID value.

Type: `bool`

Default value: `true`



**subcluster1.cpu12.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu12.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu12.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu12.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu12.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu12.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster1.cpu12.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu12.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu12.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu12.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu12.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu12.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu12.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster1.cpu12.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster1.cpu12.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: 0x123456

**subcluster1.cpu12.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster1.cpu12.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster1.cpu12.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster1.cpu12.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster1.cpu12.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: true

**subcluster1.cpu12.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu12.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu12.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster1.cpu12.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu12.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu12.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu12.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster1.cpu2.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu2.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster1.cpu2.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu2.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu2.force-fpsid`**

Override the FPSID value.

Type: `bool`

Default value: `true`

### **`subcluster1.cpu2.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu2.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu2.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu2.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu2.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to

the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu2.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

#### **`subcluster1.cpu2.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu2.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu2.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu2.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu2.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu2.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu2.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu2.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu2.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster1.cpu2.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t



Default value: 171

**subcluster1.cpu2.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu2.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu2.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu2.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu2.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu2.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: 0xf000000

**subcluster1.cpu2.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu2.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster1.cpu2.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: true

**subcluster1.cpu3.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: false

**subcluster1.cpu3.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: bool

Default value: false

**subcluster1.cpu3.CRYPTODISABLE**

Disable cryptographic features.

Type: bool

Default value: false

**subcluster1.cpu3.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu3.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu3.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster1.cpu3.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu3.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu3.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

### **`subcluster1.cpu3.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

### **`subcluster1.cpu3.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu3.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu3.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu3.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster1.cpu3.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster1.cpu3.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster1.cpu3.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster1.cpu3.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu3.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: 0x0

**subcluster1.cpu3.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: 0xf000000

**subcluster1.cpu3.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu3.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu3.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu3.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu3.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster1.cpu4.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu4.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster1.cpu4.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu4.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu4.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu4.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu4.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu4.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.



Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu4.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu4.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu4.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu4.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

### **`subcluster1.cpu4.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu4.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu4.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu4.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu4.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu4.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu4.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu4.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu4.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu4.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster1.cpu4.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster1.cpu4.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu4.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu4.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu4.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu4.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu4.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu4.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu4.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu4.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu4.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster1.cpu5.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster1.cpu5.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu5.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu5.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu5.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu5.force-fpsid`**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu5.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu5.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu5.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu5.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu5.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu5.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster1.cpu5.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu5.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu5.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu5.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu5.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu5.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu5.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu5.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu5.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu5.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster1.cpu5.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster1.cpu5.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster1.cpu5.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`



Default value: N/A

**subcluster1.cpu5.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: true

**subcluster1.cpu5.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu5.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu5.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster1.cpu5.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu5.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu5.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu5.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster1.cpu6.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu6.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster1.cpu6.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu6.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu6.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu6.force-fpsid`**

Override the FPSID value.

Type: `bool`

Default value: `true`

### **`subcluster1.cpu6.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu6.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu6.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu6.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu6.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to

the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu6.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

#### **`subcluster1.cpu6.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu6.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu6.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu6.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu6.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu6.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu6.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu6.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu6.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu6.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster1.cpu6.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster1.cpu6.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu6.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu6.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu6.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu6.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu6.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu6.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: 0xf000000

**subcluster1.cpu6.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu6.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster1.cpu6.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: true

**subcluster1.cpu7.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: false

**subcluster1.cpu7.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: bool

Default value: false

**subcluster1.cpu7.CRYPTODISABLE**

Disable cryptographic features.

Type: bool

Default value: false

**subcluster1.cpu7.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu7.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu7.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu7.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu7.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu7.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`



**subcluster1.cpu7.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu7.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu7.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster1.cpu7.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu7.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu7.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu7.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu7.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

### **`subcluster1.cpu7.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

### **`subcluster1.cpu7.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu7.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu7.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu7.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster1.cpu7.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster1.cpu7.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster1.cpu7.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster1.cpu7.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu7.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu7.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu7.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu7.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu7.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu7.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu7.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster1.cpu8.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu8.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster1.cpu8.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu8.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu8.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu8.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu8.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu8.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu8.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu8.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu8.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu8.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

### **`subcluster1.cpu8.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu8.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu8.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu8.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu8.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu8.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu8.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu8.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu8.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu8.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster1.cpu8.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster1.cpu8.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu8.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu8.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`



**subcluster1.cpu8.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu8.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu8.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster1.cpu8.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu8.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu8.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster1.cpu8.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster1.cpu9.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster1.cpu9.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu9.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu9.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu9.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu9.force-fpsid`**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu9.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster1.cpu9.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu9.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu9.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster1.cpu9.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster1.cpu9.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: 0x123456

**subcluster1.cpu9.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster1.cpu9.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster1.cpu9.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster1.cpu9.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster1.cpu9.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: true

**subcluster1.cpu9.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu9.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster1.cpu9.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu9.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu9.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu9.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster1.dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.dcache-maintenance\_latency**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.dcache-miss\_latency**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.dcache-prefetch\_enabled**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

Type: `bool`

Default value: `false`

**subcluster1.dcache-read\_access\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.dcache-size**

L1 D-Cache size in bytes.

Type: uint32\_t

Default value: 0x8000

**subcluster1.dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.dcache-write\_access\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0



**subcluster1.dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.ete.CLAIMTAGS**

Number of claim tags.

Type: uint8\_t

Default value: 4

**subcluster1.ete.ETE\_REVISION**

ETE revision: 0=ETEv1.0, 1=ETEv1.1, 2=ETEv1.2, 3=ETEv1.3.

Type: uint8\_t

Default value: 1

**subcluster1.ete.MAX\_INST\_PER\_Q**

Maximum limit for the number of instructions implied by a Q element.

Type: uint16\_t

Default value: 0x1

**subcluster1.ete.NumberOfRSPairs**

Number of resource selector pairs.

Type: uint8\_t

Default value: 8

**subcluster1.ete.PIDR\_CM0D**

TRCPIDR CM0D value.

Type: uint8\_t

Default value: 0

**subcluster1.ete.PIDR\_REVAND**

TRCPIDR REVAND value.

Type: `uint8_t`

Default value: 0

#### **`subcluster1.ete.PIDR_REVISION`**

TRCPIDR REVISION value.

Type: `uint8_t`

Default value: 0

#### **`subcluster1.ete.Q_CADENCE`**

Number of instruction blocks traced between two Q elements.

Type: `uint16_t`

Default value: 0x1

#### **`subcluster1.ete.RES0_STATEFUL`**

Whether **RES0** bits are stateful or **RAZ/WI**.

Type: `bool`

Default value: `false`

#### **`subcluster1.ete.RETSTACK`**

Return stack depth.

Type: `uint8_t`

Default value: 3

#### **`subcluster1.ete.REVISION`**

TRCIDR1 revision value.

Type: `uint8_t`

Default value: 0

#### **`subcluster1.ete.SIM_OVERFLOW_GRANULARITY`**

Number of instruction blocks in each granule, for simulated overflow.

Type: `uint32_t`

Default value: 0x64

#### **`subcluster1.ete.SIM_OVERFLOW_PERCENTAGE`**

Percentage of instruction blocks lost in each granule, for simulated overflow.

Type: `uint8_t`

Default value: 0

#### **`subcluster1.ete.SOURCE_ADDRESS`**

Allow generation of source address elements.

Type: `bool`

Default value: `false`

#### **`subcluster1.ete.TRACE_OUTPUT`**

File to which to write trace byte stream.

Type: `string`

Default value: N/A

#### **`subcluster1.ete.TRCRSRTA_FORCED_EXCEP`**

TRCRSR.TA value for a forcibly traced exception.

Type: `bool`

Default value: `false`

#### **`subcluster1.ete.TSMARK`**

Whether timestamp markers are supported.

Type: `bool`

Default value: `true`

#### **`subcluster1.force_mte_tag_access_razwi_and_ignore_tag_checks`**

Force MTE tag accesses to **RAZ/WI** and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

Type: `bool`

Default value: `false`

#### **`subcluster1.force_zero_PSTATE_PAN`**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

### **`subcluster1.force_zero_mpam_partid_and_pmg`**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAMO\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

Type: `uint8_t`

Default value: `0`

### **`subcluster1.has_delayed_dbgreg`**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

### **`subcluster1.has_delayed_sysreg`**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

### **`subcluster1.has_enhanced_pan`**

Implements Armv8.7 Enhanced PAN feature (FEAT\_PAN3) values of this parameter are:- 1, feature is implemented if ARMv8.7 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: `2`

### **`subcluster1.has_ete`**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (`-plugin` or `-P`).

Type: `bool`

Default value: `false`

### **`subcluster1.has_large_va`**

Implement support for the extended 52-bit virtual addresses from ARMv8.2 (FEAT\_LVA). values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.2 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **subcluster1.has\_statistical\_profiling**

Whether Statistical Based Profiling is implemented (FEAT\_SPE).

Type: bool

Default value: false

### **subcluster1.icache-hit\_latency**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster1.icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster1.icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster1.icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

Type: bool

Default value: false

### **subcluster1.icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.icache-read_latency`**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.icache-size`**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: `0x8000`

### **`subcluster1.instruction_tlb_size`**

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

Type: `uint32_t`

Default value: `0x0`

### **`subcluster1.invalidate_code_cache_on_icache_cmo`**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: `uint8_t`

Default value: `0`

### **`subcluster1.memory_tagging_support_level`**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

Type: `uint8_t`

Default value: `3`

### **`subcluster1.pmu-num_counters`**

Number of PMU counters implemented.

Type: `uint8_t`

Default value: 6

### **`subcluster1.ptw_latency`**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

### **`subcluster1.stage12_tlb_size`**

Number of stage1+2 tlb entries. Valid values are 0 or  $\geq 4$ .

Type: `uint32_t`

Default value: 0x80

### **`subcluster1.tlb_latency`**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

### **`subcluster1.tlbi_stall_enabled`**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: `bool`

Default value: `false`

### **`subcluster1.treat_PAC_as_NOP`**

Non-architecture parameter. Treat Pointer Authentication as **NOP**. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are **NOP**. Branch, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

Type: `bool`

Default value: `false`

### **`subcluster1.walk_cache_latency`**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

## 3.52 ARMCortexA520CT\_CortexA720CT\_CortexX4CT

Defined in LISA/ARMCortexA520CT\_CortexA720CT\_CortexX4CT.lisa.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
CortexA520 r0p1	Full support
CortexA720 r0p1	Full support
CortexX4 r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following parameters were added:

- num\_acp
- subcluster0.has\_delayed\_dbgreg
- subcluster0.has\_delayed\_sysreg
- subcluster1.has\_delayed\_dbgreg
- subcluster1.has\_delayed\_sysreg
- subcluster2.has\_delayed\_dbgreg
- subcluster2.has\_delayed\_sysreg

### About ARMCortexA520CT\_CortexA720CT\_CortexX4CT

The number of cores in each subcluster is configurable using the following parameters:

#### **subcluster0.NUM\_CORES**

Possible values are 1-12 (ARMCortexA520CT).

#### **subcluster1.NUM\_CORES**

Possible values are 1-12 (ARMCortexA720CT).

#### **subcluster2.NUM\_CORES**

Possible values are 1-12 (ARMCortexX4CT).

The total number of cores in the cluster cannot exceed 14.

Port arrays are expanded to the width needed for the maximum number of cores in each subcluster. Use the following port array indexes:

- <port\_name>[0-11] for cores in subcluster0.
- <port\_name>[12-23] for cores in subcluster1.



- `<port_name>[24-35]` for cores in `subcluster2`.

**Note**

All instances in the Master cross trigger matrix port array, `cti[36]` must be connected, regardless of the `NUM_CORES` value used.

Core-specific parameters have the following prefixes:

- `subcluster0.cpu0` to `subcluster0.cpu11` identify cores in `subcluster0`.
- `subcluster1.cpu0` to `subcluster1.cpu11` identify cores in `subcluster1`.
- `subcluster2.cpu0` to `subcluster2.cpu11` identify cores in `subcluster2`.

For information about the cores in this model, see:

- [ARMCortexA520CT](#).
- [ARMCortexA720CT](#).
- [ARMCortexX4CT](#).

## Iris and MTI instances for ARMCortexA520CT\_CortexA720CT\_CortexX4CT

This model has the following Iris instances:

Name	Instance type
ARMCortexA520CT_CortexA720CT_CortexX4CT	Cluster_ARM_CortexA520_CortexA720_CortexX4_Heterogeneous
ARMCortexA520CT_CortexA720CT_CortexX4CT.AMU	PVBusLogger
ARMCortexA520CT_CortexA720CT_CortexX4CT.AMU.mapper	PVBusMapper
ARMCortexA520CT_CortexA720CT_CortexX4CT.DAP	PVBusLogger
ARMCortexA520CT_CortexA720CT_CortexX4CT.DAP.mapper	PVBusMapper
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU	DSU-120
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.PPU_cluster	PPUv1
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.PPU_coreU (where $U = 0-2$ )	PPUv1
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.PPU_coreU.busslave (where $U = 0-2$ )	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.shared_cache	PVCache

Name	Instance type
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.shared_cache.upstream[U] (where $U = 0-8$ )	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.MMAP	PVBusLogger
ARMCortexA520CT_CortexA720CT_CortexX4CT.MMAP.mapper	PVBusMapper
ARMCortexA520CT_CortexA720CT_CortexX4CT.RAS	PVBusLogger
ARMCortexA520CT_CortexA720CT_CortexX4CT.RAS.mapper	PVBusMapper
ARMCortexA520CT_CortexA720CT_CortexX4CT.cpuU.debug_rom (where $U = 0-2$ )	debug_rom
ARMCortexA520CT_CortexA720CT_CortexX4CT.ext_bus	PVBusLogger
ARMCortexA520CT_CortexA720CT_CortexX4CT.ext_bus.mapper	PVBusMapper
ARMCortexA520CT_CortexA720CT_CortexX4CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA520CT_CortexA720CT_CortexX4CT.global_debug_rom	debug_rom
ARMCortexA520CT_CortexA720CT_CortexX4CT.secondary_debug_rom	debug_rom
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster0	Subcluster_ARM_Cortex-A520
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster0.cpu0	ARM_Cortex-A520
ARMCortexA520CT_CortexA720CT_CortexX4CT.subclusterU.cpu0.UTLB (where $U = 0-2$ )	TLB
ARMCortexA520CT_CortexA720CT_CortexX4CT.subclusterU.cpuV.dtlb (where $U = 0-2$ ; $V = 0-2$ )	TLB
ARMCortexA520CT_CortexA720CT_CortexX4CT.subclusterU.cpu0.l1dcache (where $U = 0-2$ )	PVCache
ARMCortexA520CT_CortexA720CT_CortexX4CT.subclusterU.cpu0.l1dcache.downstream[0].pvbusmaster (where $U = 0-2$ )	PVBusMaster
ARMCortexA520CT_CortexA720CT_CortexX4CT.subclusterU.cpu0.l1dcache.upstream[0] (where $U = 0-2$ )	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.subclusterU.cpu0.l1icache (where $U = 0-2$ )	PVCache
ARMCortexA520CT_CortexA720CT_CortexX4CT.subclusterU.cpu0.l1icache.downstream[0].pvbusmaster (where $U = 0-2$ )	PVBusMaster
ARMCortexA520CT_CortexA720CT_CortexX4CT.subclusterU.cpu0.l1icache.upstream[0] (where $U = 0-2$ )	PVBusSlave

Name	Instance type
ARMCortexA520CT_CortexA720CT_CortexX4CT.subclusterU.cpu0.l2cache (where $U = 0-2$ )	PVCache
ARMCortexA520CT_CortexA720CT_CortexX4CT.subclusterU.cpu0.l2cache.downstream[0].pvbusmaster (where $U = 0-2$ )	PVBusMaster
ARMCortexA520CT_CortexA720CT_CortexX4CT.subclusterU.cpu0.l2cache.upstream[A] (where $U = 0-2$ ; $A = 0-1$ )	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.subclusterU.sve (where $U = 0-2$ )	ScalableVectorExtension
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster1	Subcluster_ARM_Cortex-A720
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster1.cpu0	ARM_Cortex-A720
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster2	Subcluster_ARM_Cortex-X4
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster2.cpu0	ARM_Cortex-X4

This model has the following MTI trace components:

Name	Component type
ARMCortexA520CT_CortexA720CT_CortexX4CT.AMU	PVBusLogger
ARMCortexA520CT_CortexA720CT_CortexX4CT.AMU.mapper	PVBusMapper
ARMCortexA520CT_CortexA720CT_CortexX4CT.DAP	PVBusLogger
ARMCortexA520CT_CortexA720CT_CortexX4CT.DAP.mapper	PVBusMapper
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU	DSU-120
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.PPU_cluster	PPUv1
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.PPU_coreU (where $U = 0-2$ )	PPUv1
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.PPU_coreU.busslave (where $U = 0-2$ )	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.shared_cache	PVCache
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.shared_cache.upstream[U] (where $U = 0-8$ )	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.MMAP	PVBusLogger
ARMCortexA520CT_CortexA720CT_CortexX4CT.MMAP.mapper	PVBusMapper
ARMCortexA520CT_CortexA720CT_CortexX4CT.RAS	PVBusLogger
ARMCortexA520CT_CortexA720CT_CortexX4CT.RAS.mapper	PVBusMapper
ARMCortexA520CT_CortexA720CT_CortexX4CT.ext_bus	PVBusLogger

Name	Component type
ARMCortexA520CT_CortexA720CT_CortexX4CT.ext_bus.mapper	PVBusMapper
ARMCortexA520CT_CortexA720CT_CortexX4CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster0.cpu0	ARM_Cortex-A520
ARMCortexA520CT_CortexA720CT_CortexX4CT.subclusterU.cpu0.UTLB (where $U = 0-2$ )	TLB
ARMCortexA520CT_CortexA720CT_CortexX4CT.subclusterU.cpu0.l1dcache (where $U = 0-2$ )	PVCache
ARMCortexA520CT_CortexA720CT_CortexX4CT.subclusterU.cpu0.l1dcache.downstream[0].pvbusmaster (where $U = 0-2$ )	PVBusMaster
ARMCortexA520CT_CortexA720CT_CortexX4CT.subclusterU.cpu0.l1dcache.upstream[0] (where $U = 0-2$ )	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.subclusterU.cpu0.l1icache (where $U = 0-2$ )	PVCache
ARMCortexA520CT_CortexA720CT_CortexX4CT.subclusterU.cpu0.l1icache.downstream[0].pvbusmaster (where $U = 0-2$ )	PVBusMaster
ARMCortexA520CT_CortexA720CT_CortexX4CT.subclusterU.cpu0.l1icache.upstream[0] (where $U = 0-2$ )	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.subclusterU.cpu0.l2cache (where $U = 0-2$ )	PVCache
ARMCortexA520CT_CortexA720CT_CortexX4CT.subclusterU.cpu0.l2cache.downstream[0].pvbusmaster (where $U = 0-2$ )	PVBusMaster
ARMCortexA520CT_CortexA720CT_CortexX4CT.subclusterU.cpu0.l2cache.upstream[A] (where $U = 0-2$ ; $A = 0-1$ )	PVBusSlave
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster1.cpu0	ARM_Cortex-A720
ARMCortexA520CT_CortexA720CT_CortexX4CT.subcluster2.cpu0	ARM_Cortex-X4

### Ports for ARMCortexA520CT\_CortexA720CT\_CortexX4CT

Port	Direction	Protocol	Description
acp_s	slave	PVBus	AXI ACP slave port.
AEND0MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND2MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND3MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART2MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	slave	Signal	CHI defined pins.
broadcastcachemaint	slave	Signal	ACE defined pins.
broadcastouter	slave	Signal	ACE defined pins.

Port	Direction	Protocol	Description
broadcastpersist	slave	Signal	CHI defined pins.
cfgend	slave	Signal	This signal is for EE bit initialisation.
cfgsdisable	slave	Signal	This signal disables write access to some secure Interrupt Controller registers.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcs_m_pchannel	master	PChannel	Cluster PCSM signal
cluster_powerdown_out	master	Signal	DEPRECATED - An output from PPUs that informs thermal controller of the cluster power info
cluster_ppu_hw_stat	master	Value	Cluster PPU power state
clusterid	slave	Value	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set ManagerID64 of the core.
clusterpmuirq	master	Signal	DynamiQ pmu irq
CNTHPIRQ	master	Signal	Timer signals to SOC.
CNTHPSIRQ	master	Signal	Timer signals to SOC
CNTHVIRQ	master	Signal	Timer signals to SOC.
CNTHVSIQ	master	Signal	Timer signals to SOC
CNTPNSIRQ	master	Signal	Timer signals to SOC.
CNTPSIQ	master	Signal	Timer signals to SOC.
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.
CNTVIRQ	master	Signal	Timer signals to SOC.
commirq	master	Signal	Interrupt signal from debug communications channel.
core_clk_in	slave	ClockSignal	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
core_pcs_m_pchannel	master	PChannel	Core PCSM signals
core_powerdown_out	master	Signal	DEPRECATED - An output from PPUs that informs thermal controller of the core power info
core_ppu_hw_stat	master	Value	Notify the power state change inside the PPU.
coreerrirq	master	Signal	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq	master	Signal	Fault indicator for a detected 1 or 2 bit ECC error
coreinstrrun	master	Signal	Core Running State
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.

Port	Direction	Protocol	Description
ctidbgirq	master	Signal	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	slave	Signal	External debug interface.
dbgnopwrdown	master	Signal	No power-down request.
dbgpwrupreq	master	Signal	Debug power up request.
defaultMP	slave	Signal	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main master interface ports.
dev_debug_s	slave	PVBus	External debug interface.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.
gicreset	master	Signal	An output from PPU that can be used to reset GICCLK domain components
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	slave	PVBus	External debug interface.
pmbirq	master	Signal	Interrupt signal from the statistical profiling unit.
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	master	Signal	PPU cluster interrupt.
ppu_cluster_isolate	master	Signal	PPU Cluster Domain isolation control.
ppu_cluster_wakerequest	slave	Signal	PPU cluster wake request signal.
ppu_core_irq	master	Signal	PPU core interrupt.
ppu_core_isolate	master	Signal	PPU Core Domain isolation control.
ppu_core_wakerequest	slave	Signal	PPU core wake request signal.
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	master	PVBus	The core will generate bus requests on this port.
pvbus_periph_m	master	PVBus	The core can generate peripheral bus request on this port.
rei	slave	Signal	Per core RAM Error Interrupt.
reset	slave	Signal	Reset signal to cluster.
rvbaraddr	slave	Value_64	Reset vector base address.
sci_m	master	SystemCoherencyInterface	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei	slave	Signal	Per core System Error physical pins.
spiden	slave	Signal	External debug interface.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq	master	Signal	Interrupt signal from the trace buffer unit.
utility_bus_s	slave	PVBus	Utility bus slave

Port	Direction	Protocol	Description
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfiq	slave	Signal	Virtualised FIQ.
virq	slave	Signal	Virtualised IRQ.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Per core virtual System Error physical pins.

## Parameters for ARM CortexA520CT\_CortexA720CT\_CortexX4CT

### AEND0\_DEFAULT

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

Type: uint64\_t

Default value: 0x0

### AEND1\_DEFAULT

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

Type: uint64\_t

Default value: 0x0

### AEND2\_DEFAULT

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

Type: uint64\_t

Default value: 0x0

### AEND3\_DEFAULT

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

Type: uint64\_t

Default value: 0x0

### ASTART0\_DEFAULT

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

Type: `uint64_t`

Default value: `0x0`

### **ASTART1\_DEFAULT**

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

Type: `uint64_t`

Default value: `0x0`

### **ASTART2\_DEFAULT**

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

Type: `uint64_t`

Default value: `0x0`

### **ASTART3\_DEFAULT**

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

Type: `uint64_t`

Default value: `0x0`

### **BROADCASTATOMIC**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`

Default value: `true`

### **BROADCASTCACHEMAIN**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

Type: `bool`

Default value: `false`

### **BROADCASTOUTER**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.



Type: `bool`

Default value: `false`

### **BROADCASTPERSIST**

Enable broadcasting of cache clean to the point of persistence operations. The `broadcastpersist` signal will override this value if used.

Type: `bool`

Default value: `true`

### **CLUSTER\_ID**

The Cluster ID maps to the affinity levels of the `MPIDR_EL1` register and is evaluated based on the `MPIDR_EL1` layout. If `MPIDR_EL1` supports 16-bit cluster affinity levels, bits [15:8] map to `IDRAFF3`, while bits [7:0] map to `IDRAFF2`. If `MPIDR_EL1` supports 24-bit cluster affinity levels, the bits [23:16] map to `IDRAFF3`, bits [15:8] map to `IDRAFF2`, and bits [7:0] map to `IDRAFF1`. This configuration also updates all relevant component `DEVAFF` registers and is used to set the `ManagerID64` of the core.

Type: `uint32_t`

Default value: `0x0`

### **CMO\_broadcast\_when\_cache\_state\_modelling\_disabled**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. This value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required.

Type: `uint8_t`

Default value: `1`

### **DBGROMADDR**

Initialization value of `DBGDRAR` register. Bits[55:12] of this register specify the ROM table physical address.

Type: `uint64_t`

Default value: `0x0`

### **DBGROMADDRV**

If true, set bits[1:0] of the CP15 `DBGDRAR` to indicate that the address is valid.

Type: `bool`

Default value: `false`

### **GICDISABLE**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`

Default value: `true`

### **bus\_type**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

Type: `uint8_t`

Default value: 0

### **core\_complex\_mapping**

Defines Complex descriptions for platforms that support several Cores per Complex like Cortex-A510. JSON format: {"complex0": { "cores" : [ 0, 1 ], "l2-cache" :{"exists":1, "size":16MB}}, ... , "complexN": { "cores" : [<core\_list>], "l2-cache" : {"exists":1, "size":16MB}}} where <core\_list> is the list of cores in the complexN. Effective only when the parameter value is not empty.

Type: `string`

Default value: {"complex0": { "cores": [0, 1], "l2-cache" :{"exists":1, "size":"16MB"}}, "complex1": { "cores": [2, 3], "l2-cache" :{"exists":1, "size":"16MB"}}, "complex2": { "cores": [4, 5], "l2-cache" :{"exists":1, "size":"16MB"}}, "complex3": { "cores": [6, 7], "l2-cache" :{"exists":1, "size":"16MB"}}, "complex4": { "cores": [8, 9], "l2-cache" :{"exists":1, "size":"16MB"}}, "complex5": { "cores": [10, 11], "l2-cache" :{"exists":1, "size":"16MB"}}, "complex6": { "cores": [12, 13], "l2-cache" :{"exists":1, "size":"16MB"}} }

### **core\_power\_on\_by\_default**

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPUs which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

Type: `bool`

Default value: `false`

### **dcache-state\_modelled**

Set whether D-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **`enable_simulation_performance_optimizations`**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of `stage12_tlb_size` parameter to 1024).

Type: `bool`

Default value: `true`

### **`has_peripheral_port`**

If `true`, an additional AXI peripheral port is configured.

Type: `bool`

Default value: `false`

### **`icache-state_modelled`**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **`l3cache-has_mpam`**

L3 Cache has MPAM support.

Type: `bool`

Default value: `true`

### **`l3cache-hit_latency`**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-maintenance_latency`**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-miss\_latency**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-read\_access\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size `l3cache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-read\_bus\_width\_in\_bytes**

L3 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: `uint32_t`

Default value: `0x10`

**l3cache-read\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l3cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-size**

L3 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

**l3cache-snoop\_data\_transfer\_latency**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **l3cache-snoop\_issue\_latency**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **l3cache-ways**

L3 Cache number of ways (sets are implicit from size).

Type: uint16\_t

Default value: 16

### **l3cache-write\_access\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size l3cache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **l3cache-write\_bus\_width\_in\_bytes**

L3 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

### **l3cache-write\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. l3cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **mpmm\_accumulator\_multiplier**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator.value of n means the accumulator will use (n \* accumulator value) to calculate the mpmm threshold (MPMM).is provided as a fast model workaround to handle cases where execution of quantums

in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

Type: `uint8_t`

Default value: 1

### **num\_acp**

Number of ACP ports.

Type: `uint8_t`

Default value: 0

### **num\_nodes**

Number of transport nodes. Zero implies direct-connect configuration.

Type: `uint8_t`

Default value: 1

### **revision\_number**

Revision number of TRM implemented by the model. Changing the value can change the model behaviour. Visible as the Variant field in MIDR/MIDR\_EL1. Corresponds to the revision number X in rXpY.

Type: `uint8_t`

Default value: 0

### **subcluster0.CPUCFR**

Value of CPU Configuration Register.

Type: `uint64_t`

Default value: 0x0

### **subcluster0.NUM\_CORES**

Number of cores in the subcluster. Total number of cores in cluster may not exceed 14.

Type: `uint8_t`

Default value: 1

### **subcluster0.core\_cache\_protection**

`core_cache_protection` can change `ERROFR`, `ERROPFGF` and `ERROPFGCTL` fields. Possible values are: -1:Not implemented (by default), 0:Disabled, 1:Enabled.

Type: `int8_t`

Default value: 1

### **`subcluster0.cpi_div`**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: 0x1

### **`subcluster0.cpi_mul`**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: 0x1

### **`subcluster0.cpu0.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu0.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu0.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: 0x0

### **`subcluster0.cpu0.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu0.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu0.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t



Default value: 0x0

**subcluster0.cpu0.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu0.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu0.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu0.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu0.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu0.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu0.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu0.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu0.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

### **`subcluster0.cpu0.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

### **`subcluster0.cpu0.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **`subcluster0.cpu0.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **`subcluster0.cpu0.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster0.cpu0.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu0.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu0.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster0.cpu0.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu0.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu0.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu0.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu1.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu1.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster0.cpu1.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu1.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu1.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu1.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu1.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu1.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster0.cpu1.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu1.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu1.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu1.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu1.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu1.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu1.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu1.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu1.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu1.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster0.cpu1.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster0.cpu1.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu1.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu1.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`



**subcluster0.cpu1.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu1.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster0.cpu1.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu1.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu1.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster0.cpu1.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster0.cpu10.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu10.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu10.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu10.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu10.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu10.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu10.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu10.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu10.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu10.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu10.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu10.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu10.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu10.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu10.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu10.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu10.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu10.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu10.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu10.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu10.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu10.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu10.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster0.cpu10.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster0.cpu10.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster0.cpu10.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster0.cpu10.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu10.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu10.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu10.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu10.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu10.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster0.cpu10.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu10.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster0.cpu11.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu11.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu11.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu11.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu11.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu11.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu11.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu11.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu11.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu11.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t



Default value: 0x0

#### **subcluster0.cpu11.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

#### **subcluster0.cpu11.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu11.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu11.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

#### **subcluster0.cpu11.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu11.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu11.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu11.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu11.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu11.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu11.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu11.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu11.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**`subcluster0.cpu11.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**`subcluster0.cpu11.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**`subcluster0.cpu11.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**`subcluster0.cpu11.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**`subcluster0.cpu11.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**`subcluster0.cpu11.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**`subcluster0.cpu11.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster0.cpu11.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu11.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu11.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu11.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu2.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu2.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster0.cpu2.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu2.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu2.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu2.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu2.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu2.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster0.cpu2.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu2.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu2.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu2.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu2.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu2.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu2.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu2.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu2.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu2.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster0.cpu2.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster0.cpu2.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu2.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu2.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`



**subcluster0.cpu2.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu2.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu2.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu2.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu2.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu2.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster0.cpu2.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu3.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu3.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu3.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu3.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu3.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu3.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu3.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu3.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu3.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu3.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu3.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu3.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu3.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu3.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu3.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu3.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster0.cpu3.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster0.cpu3.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster0.cpu3.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster0.cpu3.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu3.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu3.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu3.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu3.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu3.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster0.cpu3.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu3.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster0.cpu4.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu4.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu4.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu4.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu4.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu4.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t



Default value: 0x0

**subcluster0.cpu4.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu4.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu4.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu4.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu4.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu4.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu4.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu4.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu4.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

### **`subcluster0.cpu4.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

### **`subcluster0.cpu4.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **`subcluster0.cpu4.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **`subcluster0.cpu4.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster0.cpu4.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu4.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu4.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster0.cpu4.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu4.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu4.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu4.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu5.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu5.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster0.cpu5.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu5.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu5.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu5.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu5.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu5.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster0.cpu5.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu5.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu5.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu5.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu5.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu5.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu5.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu5.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu5.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu5.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster0.cpu5.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster0.cpu5.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu5.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu5.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`



**subcluster0.cpu5.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu5.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster0.cpu5.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu5.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu5.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster0.cpu5.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu6.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu6.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu6.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu6.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu6.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu6.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu6.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu6.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu6.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu6.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu6.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu6.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu6.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu6.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu6.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu6.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster0.cpu6.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster0.cpu6.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster0.cpu6.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster0.cpu6.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu6.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu6.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu6.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu6.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu6.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster0.cpu6.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu6.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster0.cpu7.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu7.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu7.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu7.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu7.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu7.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t



Default value: 0x0

**subcluster0.cpu7.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu7.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu7.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu7.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu7.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu7.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu7.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu7.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu7.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

### **`subcluster0.cpu7.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

### **`subcluster0.cpu7.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **`subcluster0.cpu7.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **`subcluster0.cpu7.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster0.cpu7.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu7.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu7.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster0.cpu7.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu7.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu7.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu7.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu8.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu8.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu8.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu8.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu8.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu8.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu8.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu8.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster0.cpu8.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu8.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu8.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu8.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu8.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu8.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu8.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu8.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu8.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu8.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster0.cpu8.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster0.cpu8.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu8.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu8.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`



**subcluster0.cpu8.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu8.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster0.cpu8.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu8.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu8.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster0.cpu8.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu9.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu9.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu9.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu9.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu9.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu9.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu9.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu9.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu9.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu9.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu9.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu9.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu9.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu9.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu9.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu9.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster0.cpu9.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster0.cpu9.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster0.cpu9.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster0.cpu9.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu9.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu9.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu9.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu9.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu9.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster0.cpu9.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu9.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster0.dcache-hit_latency`**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.dcache-maintenance_latency`**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.dcache-miss_latency`**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.dcache-prefetch_enabled`**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

Type: `bool`

Default value: `false`

### **`subcluster0.dcache-read_access_latency`**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.dcache-read_latency`**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.dcache-size`**

L1 D-Cache size in bytes.

Type: `uint32_t`

Default value: `0x8000`

### **`subcluster0.dcache-snoop_data_transfer_latency`**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.dcache-write_access_latency`**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`



**subcluster0.dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.ete.CLAIMTAGS**

Number of claim tags.

Type: uint8\_t

Default value: 4

**subcluster0.ete.MAX\_INST\_PER\_Q**

Maximum limit for the number of instructions implied by a Q element.

Type: uint16\_t

Default value: 0x1

**subcluster0.ete.PIDR\_CM0D**

TRCPIDR CM0D value.

Type: uint8\_t

Default value: 0

**subcluster0.ete.PIDR\_REVAND**

TRCPIDR REVAND value.

Type: uint8\_t

Default value: 0

**subcluster0.ete.PIDR\_REVISION**

TRCPIDR REVISION value.

Type: uint8\_t

Default value: 0

**subcluster0.ete.Q\_CADENCE**

Number of instruction blocks traced between two Q elements.

Type: `uint16_t`

Default value: `0x1`

#### **`subcluster0.ete.RES0_STATEFUL`**

Whether **RES0** bits are stateful or **RAZ/WI**.

Type: `bool`

Default value: `false`

#### **`subcluster0.ete.RETSTACK`**

Return stack depth.

Type: `uint8_t`

Default value: `3`

#### **`subcluster0.ete.REVISION`**

TRCIDR1 revision value.

Type: `uint8_t`

Default value: `0`

#### **`subcluster0.ete.SIM_OVERFLOW_GRANULARITY`**

Number of instruction blocks in each granule, for simulated overflow.

Type: `uint32_t`

Default value: `0x64`

#### **`subcluster0.ete.SIM_OVERFLOW_PERCENTAGE`**

Percentage of instruction blocks lost in each granule, for simulated overflow.

Type: `uint8_t`

Default value: `0`

#### **`subcluster0.ete.SOURCE_ADDRESS`**

Allow generation of source address elements.

Type: `bool`

Default value: `false`

#### **`subcluster0.ete.TRACE_OUTPUT`**

File to which to write trace byte stream.

Type: `string`

Default value: `N/A`

#### **`subcluster0.ete.TRCRSRTA_FORCED_EXCEP`**

TRCRSR.TA value for a forcibly traced exception.

Type: `bool`

Default value: `false`

#### **`subcluster0.force_mte_tag_access_razwi_and_ignore_tag_checks`**

Force MTE tag accesses to **RAZ/WI** and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

Type: `bool`

Default value: `false`

#### **`subcluster0.force_zero_PSTATE_PAN`**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

#### **`subcluster0.force_zero_mpam_partid_and_pmg`**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

Type: `uint8_t`

Default value: `0`

#### **`subcluster0.has_actlr2`**

If true ACLTR2 exists and ACTLR2(NS) is aliased to ACTLR\_EL1[63:32].

Type: `bool`

Default value: `true`

#### **`subcluster0.has_delayed_dbgreg`**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

### **subcluster0.has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

### **subcluster0.has\_dot\_product**

Possible values of this parameter are:- 1, feature not implemented.- 2, feature is implemented.

Type: `uint8_t`

Default value: 2

### **subcluster0.has\_ete**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (-plugin or -P).

Type: `bool`

Default value: `false`

### **subcluster0.icache-hit\_latency**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **subcluster0.icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **subcluster0.icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **subcluster0.icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

Type: bool

Default value: false

### **subcluster0.icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster0.icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster0.icache-size**

L1 I-Cache size in bytes.

Type: uint32\_t

Default value: 0x8000

### **subcluster0.invalidate\_code\_cache\_on\_icache\_cmo**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: uint8\_t

Default value: 0

**subcluster0.memory\_tagging\_support\_level**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

Type: uint8\_t

Default value: 3

**subcluster0.pmu-num\_counters**

Number of PMU counters implemented.

Type: uint8\_t

Default value: 6

**subcluster0.ptw\_latency**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

**subcluster0.ras\_extra\_configurations**

Miscellaneous configurations for error records. An array of JSON objects. Note for ERXCTLR\_EL1 register it only allows to define the mask value for the IMPDEF fields, ie bits [63:32] and bit 1, but its reset value applies on all fields. Note for ERXMISCN masks - these are 64 bit masks covering the 64 bit registers ERXMISCN\_EL1. E.g. [{"Index": 0, "ERXMISCO\_mask": 0x0, "ERXMISCO\_reset": 0x0, "ERXMISC1\_mask": 0x0, "ERXMISC1\_reset": 0x0, "ERXMISC2\_mask": 0x0, "ERXMISC2\_reset": 0x0, "ERXMISC3\_mask": 0x0, "ERXMISC3\_reset": 0x0, "ERXCTLR\_EL1\_mask": 0x0, "ERXCTLR\_EL1\_reset": 0x0}, {"Index": 1, "ERXMISCO\_mask": 0x0, "ERXMISCO\_reset": 0x0, "ERXSTATUS\_IERR\_mask": 0x300}].

Type: string

Default value: "[ { \"Index\": 1, \"ERXMISCO\_mask\": 0xFFFFc0003fc3, \"ERXMISC1\_mask\": 0x03F870003FF30f07, \"ERXPFGCTL\_reset\": 0x1000 }, { \"Index\": 2, \"ERXMISCO\_mask\": 0xFFFFFe007ffc0, \"ERXMISCO\_reset\": 0x2, \"ERXSTATUS\_IERR\_mask\": 0x300 , \"ERXMISC1\_mask\": 0x0FF8700ffff31f0f, \"ERXPFGCTL\_reset\": 0x1000 } ]"

**subcluster0.ras\_pfg\_clock\_mhz**

RAS Pseudo-Fault generation clock rate in MHz.

Type: uint8\_t

Default value: 12

**subcluster0.revision\_number**

Revision number of TRM implemented by the model. Changing the value can change the model behaviour. Visible as the Variant field in MIDR/MIDR\_EL1. Corresponds to the revision number X in rXpY.

Type: `uint8_t`

Default value: 0

**subcluster0.tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

**subcluster0.tlbi\_stall\_enabled**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: `bool`

Default value: `false`

**subcluster0.treat\_PAC\_as\_NOP**

Non-architecture parameter. Treat Pointer Authentication as **NOP**. When the parameter is true behaviour of FEAT\_PAAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are **NOP**. Branch, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAAuth traps.

Type: `bool`

Default value: `false`

**subcluster0.walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

**subcluster1.CPUCFR**

Value of CPU Configuration Register.

Type: `uint64_t`

Default value: 0x0

**subcluster1.NUM\_CORES**

Number of cores in the subcluster. Total number of cores in cluster may not exceed 14.

Type: `uint8_t`

Default value: 1

**subcluster1.cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: 0x1

**subcluster1.cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: 0x1

**subcluster1.cpu0.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu0.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster1.cpu0.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu0.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`



Default value: 0x0

**subcluster1.cpu0.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter trace\_special\_hlt\_imm16.

Type: bool

Default value: false

**subcluster1.cpu0.force-fpsid**

Override the FPSID value.

Type: bool

Default value: true

**subcluster1.cpu0.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu0.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu0.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu0.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu0.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu0.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

#### **subcluster1.cpu0.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu0.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu0.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu0.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu0.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

### **`subcluster1.cpu0.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

### **`subcluster1.cpu0.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu0.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu0.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

### **`subcluster1.cpu0.semihosting-T32_HLT`**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster1.cpu0.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster1.cpu0.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster1.cpu0.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster1.cpu0.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu0.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu0.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf0000000`

**subcluster1.cpu0.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu0.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu0.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu0.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu0.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster1.cpu1.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu1.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster1.cpu1.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu1.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu1.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu1.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu1.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu1.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu1.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu1.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu1.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu1.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

#### **`subcluster1.cpu1.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu1.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu1.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu1.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu1.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu1.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu1.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu1.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000



**subcluster1.cpu1.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu1.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster1.cpu1.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster1.cpu1.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu1.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu1.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu1.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu1.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu1.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu1.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu1.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu1.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu1.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster1.cpu10.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu10.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster1.cpu10.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu10.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu10.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu10.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu10.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu10.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu10.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu10.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu10.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu10.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

**subcluster1.cpu10.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu10.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu10.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu10.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu10.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu10.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu10.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu10.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu10.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu10.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster1.cpu10.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster1.cpu10.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu10.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu10.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu10.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu10.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu10.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu10.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu10.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu10.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu10.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster1.cpu11.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster1.cpu11.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu11.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu11.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu11.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`



**subcluster1.cpu11.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu11.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu11.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu11.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu11.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu11.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

#### **subcluster1.cpu11.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

#### **subcluster1.cpu11.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu11.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu11.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu11.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu11.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu11.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu11.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu11.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu11.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu11.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster1.cpu11.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster1.cpu11.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster1.cpu11.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster1.cpu11.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu11.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu11.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu11.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu11.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu11.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster1.cpu11.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu11.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster1.cpu2.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu2.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu2.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu2.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter trace\_special\_hlt\_imm16.

Type: `bool`

Default value: `false`

**subcluster1.cpu2.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu2.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

**subcluster1.cpu2.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster1.cpu2.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu2.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu2.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu2.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu2.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu2.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster1.cpu2.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.



Type: `uint8_t`

Default value: 171

### **`subcluster1.cpu2.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **`subcluster1.cpu2.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **`subcluster1.cpu2.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster1.cpu2.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu2.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster1.cpu2.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster1.cpu2.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

#### **`subcluster1.cpu2.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster1.cpu2.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu2.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster1.cpu3.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu3.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu3.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu3.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu3.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu3.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster1.cpu3.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu3.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu3.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

### **`subcluster1.cpu3.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

### **`subcluster1.cpu3.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu3.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu3.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu3.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster1.cpu3.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster1.cpu3.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster1.cpu3.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster1.cpu3.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu3.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: 0x0

**subcluster1.cpu3.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: 0xf000000

**subcluster1.cpu3.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu3.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu3.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu3.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu3.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster1.cpu4.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu4.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster1.cpu4.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu4.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu4.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu4.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu4.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu4.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.



Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu4.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu4.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu4.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu4.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

### **`subcluster1.cpu4.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu4.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu4.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu4.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu4.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu4.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu4.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu4.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu4.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu4.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster1.cpu4.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster1.cpu4.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu4.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu4.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu4.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu4.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu4.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu4.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu4.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu4.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu4.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster1.cpu5.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu5.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu5.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu5.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu5.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu5.force-fpsid`**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu5.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu5.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu5.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu5.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu5.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu5.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster1.cpu5.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu5.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu5.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu5.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu5.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu5.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu5.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster1.cpu5.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster1.cpu5.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: 0x123456

**subcluster1.cpu5.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster1.cpu5.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster1.cpu5.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster1.cpu5.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`



Default value: N/A

**subcluster1.cpu5.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: true

**subcluster1.cpu5.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu5.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu5.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster1.cpu5.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu5.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu5.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu5.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster1.cpu6.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu6.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster1.cpu6.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu6.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu6.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu6.force-fpsid`**

Override the FPSID value.

Type: `bool`

Default value: `true`

### **`subcluster1.cpu6.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu6.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu6.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu6.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu6.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to

the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu6.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

#### **`subcluster1.cpu6.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu6.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu6.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu6.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu6.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu6.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu6.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu6.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu6.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu6.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster1.cpu6.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster1.cpu6.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu6.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu6.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu6.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu6.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu6.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu6.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: 0xf000000

**subcluster1.cpu6.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu6.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster1.cpu6.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: true

**subcluster1.cpu7.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: false

**subcluster1.cpu7.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: bool

Default value: false

**subcluster1.cpu7.CRYPTODISABLE**

Disable cryptographic features.

Type: bool

Default value: false

**subcluster1.cpu7.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu7.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu7.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu7.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu7.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu7.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`



**subcluster1.cpu7.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu7.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu7.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster1.cpu7.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu7.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu7.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu7.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu7.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

### **`subcluster1.cpu7.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

### **`subcluster1.cpu7.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu7.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu7.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu7.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster1.cpu7.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster1.cpu7.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster1.cpu7.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster1.cpu7.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu7.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: 0x0

**subcluster1.cpu7.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: 0xf000000

**subcluster1.cpu7.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu7.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu7.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu7.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu7.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster1.cpu8.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu8.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster1.cpu8.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu8.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu8.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu8.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu8.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu8.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu8.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu8.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu8.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu8.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

### **`subcluster1.cpu8.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu8.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu8.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu8.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu8.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu8.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu8.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu8.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu8.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu8.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster1.cpu8.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster1.cpu8.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu8.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu8.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`



**subcluster1.cpu8.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu8.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu8.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster1.cpu8.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu8.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu8.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster1.cpu8.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster1.cpu9.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster1.cpu9.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu9.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu9.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu9.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu9.force-fpsid`**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu9.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster1.cpu9.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu9.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu9.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster1.cpu9.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster1.cpu9.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: 0x123456

**subcluster1.cpu9.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster1.cpu9.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster1.cpu9.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster1.cpu9.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster1.cpu9.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: true

**subcluster1.cpu9.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu9.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster1.cpu9.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu9.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu9.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu9.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster1.dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.dcache-maintenance\_latency**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.dcache-miss\_latency**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.dcache-prefetch\_enabled**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

Type: `bool`

Default value: `false`

**subcluster1.dcache-read\_access\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.dcache-size**

L1 D-Cache size in bytes.

Type: uint32\_t

Default value: 0x8000

**subcluster1.dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.dcache-write\_access\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0



**subcluster1.dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.ete.CLAIMTAGS**

Number of claim tags.

Type: uint8\_t

Default value: 4

**subcluster1.ete.ETE\_REVISION**

ETE revision: 0=ETEv1.0, 1=ETEv1.1, 2=ETEv1.2, 3=ETEv1.3.

Type: uint8\_t

Default value: 1

**subcluster1.ete.MAX\_INST\_PER\_Q**

Maximum limit for the number of instructions implied by a Q element.

Type: uint16\_t

Default value: 0x1

**subcluster1.ete.NumberOfRSPairs**

Number of resource selector pairs.

Type: uint8\_t

Default value: 8

**subcluster1.ete.PIDR\_CM0D**

TRCPIDR CM0D value.

Type: uint8\_t

Default value: 0

**subcluster1.ete.PIDR\_REVAND**

TRCPIDR REVAND value.

Type: `uint8_t`

Default value: 0

#### **subcluster1.ete.PIDR\_REVISION**

TRCPIDR REVISION value.

Type: `uint8_t`

Default value: 0

#### **subcluster1.ete.Q\_CADENCE**

Number of instruction blocks traced between two Q elements.

Type: `uint16_t`

Default value: 0x1

#### **subcluster1.ete.RES0\_STATEFUL**

Whether **RES0** bits are stateful or **RAZ/WI**.

Type: `bool`

Default value: `false`

#### **subcluster1.ete.RETSTACK**

Return stack depth.

Type: `uint8_t`

Default value: 3

#### **subcluster1.ete.REVISION**

TRCIDR1 revision value.

Type: `uint8_t`

Default value: 0

#### **subcluster1.ete.SIM\_OVERFLOW\_GRANULARITY**

Number of instruction blocks in each granule, for simulated overflow.

Type: `uint32_t`

Default value: 0x64

#### **subcluster1.ete.SIM\_OVERFLOW\_PERCENTAGE**

Percentage of instruction blocks lost in each granule, for simulated overflow.

Type: `uint8_t`

Default value: 0

#### **`subcluster1.ete.SOURCE_ADDRESS`**

Allow generation of source address elements.

Type: `bool`

Default value: `false`

#### **`subcluster1.ete.TRACE_OUTPUT`**

File to which to write trace byte stream.

Type: `string`

Default value: N/A

#### **`subcluster1.ete.TRCRSRTA_FORCED_EXCEP`**

TRCRSR.TA value for a forcibly traced exception.

Type: `bool`

Default value: `false`

#### **`subcluster1.ete.TSMARK`**

Whether timestamp markers are supported.

Type: `bool`

Default value: `true`

#### **`subcluster1.force_mte_tag_access_razwi_and_ignore_tag_checks`**

Force MTE tag accesses to **RAZ/WI** and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

Type: `bool`

Default value: `false`

#### **`subcluster1.force_zero_PSTATE_PAN`**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

### **`subcluster1.force_zero_mpam_partid_and_pmg`**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAMO\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

Type: `uint8_t`

Default value: `0`

### **`subcluster1.has_delayed_dbgreg`**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

### **`subcluster1.has_delayed_sysreg`**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

### **`subcluster1.has_enhanced_pan`**

Implements Armv8.7 Enhanced PAN feature (FEAT\_PAN3) values of this parameter are:- 1, feature is implemented if ARMv8.7 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: `2`

### **`subcluster1.has_ete`**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (`-plugin` or `-P`).

Type: `bool`

Default value: `false`

### **`subcluster1.has_large_va`**

Implement support for the extended 52-bit virtual addresses from ARMv8.2 (FEAT\_LVA). values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.2 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **subcluster1.has\_statistical\_profiling**

Whether Statistical Based Profiling is implemented (FEAT\_SPE).

Type: `bool`

Default value: `false`

### **subcluster1.icache-hit\_latency**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **subcluster1.icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **subcluster1.icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **subcluster1.icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

### **subcluster1.icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.icache-read_latency`**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.icache-size`**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: `0x8000`

#### **`subcluster1.instruction_tlb_size`**

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

Type: `uint32_t`

Default value: `0x0`

#### **`subcluster1.invalidate_code_cache_on_icache_cmo`**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: `uint8_t`

Default value: `0`

#### **`subcluster1.memory_tagging_support_level`**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

Type: `uint8_t`

Default value: `3`

#### **`subcluster1.pmu-num_counters`**

Number of PMU counters implemented.

Type: `uint8_t`

Default value: 6

### **`subcluster1.ptw_latency`**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

### **`subcluster1.stage12_tlb_size`**

Number of stage1+2 tlb entries. Valid values are 0 or  $\geq 4$ .

Type: `uint32_t`

Default value: 0x80

### **`subcluster1.tlb_latency`**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

### **`subcluster1.tlbi_stall_enabled`**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: `bool`

Default value: `false`

### **`subcluster1.treat_PAC_as_NOP`**

Non-architecture parameter. Treat Pointer Authentication as **NOP**. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are **NOP**. Branch, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

Type: `bool`

Default value: `false`

### **`subcluster1.walk_cache_latency`**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

#### **subcluster2.CPUCFR**

Value of CPU Configuration Register.

Type: uint64\_t

Default value: 0x0

#### **subcluster2.NUM\_CORES**

Number of cores in the subcluster. Total number of cores in cluster may not exceed 14.

Type: uint8\_t

Default value: 1

#### **subcluster2.cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: uint32\_t

Default value: 0x1

#### **subcluster2.cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: uint32\_t

Default value: 0x1

#### **subcluster2.cpu0.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: false

#### **subcluster2.cpu0.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: bool

Default value: false

#### **subcluster2.cpu0.CRYPTODISABLE**

Disable cryptographic features.



Type: `bool`

Default value: `false`

### **`subcluster2.cpu0.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu0.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu0.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu0.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu0.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu0.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu0.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu0.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

### **`subcluster2.cpu0.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu0.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu0.l2cache-ways`**

L2 Cache number of ways (sets are implicit from size).

Type: `uint8_t`

Default value: `8`

### **`subcluster2.cpu0.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu0.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu0.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

### **`subcluster2.cpu0.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

### **`subcluster2.cpu0.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster2.cpu0.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster2.cpu0.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster2.cpu0.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster2.cpu0.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster2.cpu0.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster2.cpu0.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster2.cpu0.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster2.cpu0.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu0.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu0.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster2.cpu0.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu0.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster2.cpu0.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster2.cpu0.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster2.cpu1.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster2.cpu1.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster2.cpu1.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster2.cpu1.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu1.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster2.cpu1.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu1.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu1.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster2.cpu1.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster2.cpu1.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster2.cpu1.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

#### **`subcluster2.cpu1.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster2.cpu1.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu1.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: `uint8_t`

Default value: 8

**subcluster2.cpu1.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster2.cpu1.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster2.cpu1.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: 0x100

**subcluster2.cpu1.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: 0

**subcluster2.cpu1.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000



**subcluster2.cpu1.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster2.cpu1.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster2.cpu1.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster2.cpu1.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster2.cpu1.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster2.cpu1.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster2.cpu1.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster2.cpu1.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu1.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster2.cpu1.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster2.cpu1.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster2.cpu1.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu1.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster2.cpu1.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster2.cpu10.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster2.cpu10.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

#### **`subcluster2.cpu10.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster2.cpu10.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster2.cpu10.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

#### **`subcluster2.cpu10.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu10.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu10.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu10.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu10.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu10.l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

**subcluster2.cpu10.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster2.cpu10.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu10.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster2.cpu10.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu10.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu10.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster2.cpu10.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster2.cpu10.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster2.cpu10.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster2.cpu10.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: 0x123456

**subcluster2.cpu10.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster2.cpu10.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster2.cpu10.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster2.cpu10.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster2.cpu10.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: true

**subcluster2.cpu10.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu10.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster2.cpu10.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster2.cpu10.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster2.cpu10.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu10.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster2.cpu10.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster2.cpu11.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster2.cpu11.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster2.cpu11.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster2.cpu11.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu11.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.



Type: `bool`

Default value: `false`

### **`subcluster2.cpu11.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu11.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu11.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu11.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu11.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu11.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster2.cpu11.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu11.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu11.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster2.cpu11.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu11.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu11.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster2.cpu11.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster2.cpu11.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu11.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu11.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster2.cpu11.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster2.cpu11.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster2.cpu11.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster2.cpu11.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster2.cpu11.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster2.cpu11.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu11.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu11.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster2.cpu11.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: 0xf000000

**subcluster2.cpu11.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu11.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster2.cpu11.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: true

**subcluster2.cpu2.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: false

**subcluster2.cpu2.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: bool

Default value: false

**subcluster2.cpu2.CRYPTODISABLE**

Disable cryptographic features.

Type: bool

Default value: false

**subcluster2.cpu2.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu2.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster2.cpu2.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu2.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu2.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu2.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu2.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu2.l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

**subcluster2.cpu2.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu2.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu2.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: `uint8_t`

Default value: `8`

**subcluster2.cpu2.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu2.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu2.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster2.cpu2.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster2.cpu2.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu2.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu2.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster2.cpu2.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.



Type: `uint8_t`

Default value: 60

### **`subcluster2.cpu2.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

### **`subcluster2.cpu2.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **`subcluster2.cpu2.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **`subcluster2.cpu2.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster2.cpu2.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu2.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster2.cpu2.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster2.cpu2.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster2.cpu2.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster2.cpu2.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu2.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster2.cpu3.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster2.cpu3.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu3.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu3.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu3.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu3.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu3.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu3.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu3.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu3.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu3.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster2.cpu3.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu3.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu3.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster2.cpu3.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu3.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu3.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster2.cpu3.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster2.cpu3.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu3.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu3.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster2.cpu3.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster2.cpu3.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster2.cpu3.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster2.cpu3.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster2.cpu3.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster2.cpu3.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu3.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu3.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster2.cpu3.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu3.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster2.cpu3.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster2.cpu3.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster2.cpu4.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster2.cpu4.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster2.cpu4.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster2.cpu4.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu4.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster2.cpu4.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu4.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.



Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu4.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu4.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu4.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu4.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

### **`subcluster2.cpu4.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu4.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu4.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster2.cpu4.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu4.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu4.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster2.cpu4.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster2.cpu4.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster2.cpu4.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster2.cpu4.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster2.cpu4.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster2.cpu4.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster2.cpu4.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster2.cpu4.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster2.cpu4.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster2.cpu4.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu4.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu4.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster2.cpu4.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu4.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster2.cpu4.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu4.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster2.cpu5.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster2.cpu5.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu5.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu5.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu5.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster2.cpu5.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu5.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu5.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu5.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu5.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu5.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

#### **subcluster2.cpu5.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster2.cpu5.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster2.cpu5.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

#### **subcluster2.cpu5.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster2.cpu5.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster2.cpu5.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster2.cpu5.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster2.cpu5.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu5.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu5.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster2.cpu5.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster2.cpu5.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster2.cpu5.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string



Default value: N/A

**subcluster2.cpu5.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster2.cpu5.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster2.cpu5.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu5.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu5.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster2.cpu5.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu5.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster2.cpu5.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster2.cpu5.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster2.cpu6.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster2.cpu6.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

#### **`subcluster2.cpu6.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster2.cpu6.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu6.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter trace\_special\_hlt\_imm16.

Type: `bool`

Default value: `false`

**subcluster2.cpu6.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu6.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu6.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu6.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu6.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: 0x0

**subcluster2.cpu6.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster2.cpu6.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu6.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu6.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster2.cpu6.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu6.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu6.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster2.cpu6.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster2.cpu6.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu6.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu6.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster2.cpu6.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster2.cpu6.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

### **subcluster2.cpu6.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **subcluster2.cpu6.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **subcluster2.cpu6.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **subcluster2.cpu6.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **subcluster2.cpu6.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **subcluster2.cpu6.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **subcluster2.cpu6.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster2.cpu6.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster2.cpu6.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu6.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster2.cpu7.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster2.cpu7.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu7.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu7.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu7.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu7.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu7.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu7.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu7.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`



Default value: 0x0

### **subcluster2.cpu7.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster2.cpu7.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

### **subcluster2.cpu7.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster2.cpu7.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster2.cpu7.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

### **subcluster2.cpu7.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster2.cpu7.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster2.cpu7.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

### **subcluster2.cpu7.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

### **subcluster2.cpu7.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

### **subcluster2.cpu7.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

### **subcluster2.cpu7.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster2.cpu7.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster2.cpu7.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster2.cpu7.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster2.cpu7.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster2.cpu7.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster2.cpu7.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu7.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu7.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster2.cpu7.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu7.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster2.cpu7.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster2.cpu7.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster2.cpu8.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster2.cpu8.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster2.cpu8.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster2.cpu8.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu8.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster2.cpu8.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu8.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu8.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu8.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu8.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu8.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

### **`subcluster2.cpu8.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu8.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu8.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: `uint8_t`

Default value: 8

**subcluster2.cpu8.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster2.cpu8.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster2.cpu8.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: 0x100

**subcluster2.cpu8.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: 0

**subcluster2.cpu8.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster2.cpu8.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster2.cpu8.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster2.cpu8.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster2.cpu8.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster2.cpu8.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster2.cpu8.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster2.cpu8.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`



**subcluster2.cpu8.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu8.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu8.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster2.cpu8.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu8.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster2.cpu8.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster2.cpu8.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **subcluster2.cpu9.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **subcluster2.cpu9.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **subcluster2.cpu9.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **subcluster2.cpu9.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **subcluster2.cpu9.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **subcluster2.cpu9.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu9.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu9.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu9.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu9.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu9.l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

**subcluster2.cpu9.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster2.cpu9.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu9.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster2.cpu9.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu9.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu9.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster2.cpu9.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster2.cpu9.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster2.cpu9.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster2.cpu9.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: 0x123456

**subcluster2.cpu9.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster2.cpu9.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster2.cpu9.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster2.cpu9.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster2.cpu9.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: true

**subcluster2.cpu9.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu9.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster2.cpu9.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster2.cpu9.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster2.cpu9.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu9.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster2.cpu9.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster2.dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.dcache-maintenance\_latency**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.dcache-miss\_latency**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.dcache-prefetch\_enabled**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

Type: `bool`

Default value: `false`

**subcluster2.dcache-read\_access\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.dcache-size**

L1 D-Cache size in bytes.

Type: uint32\_t

Default value: 0x10000

**subcluster2.dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.dcache-write\_access\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0



**subcluster2.dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.ecv\_support\_level**

Implement Enhanced Counter Virtualization feature from ARMv8.6. 0, Not supported. 1, fully supported without CNTPOFF. 2, fully supported with CNTPOFF (FEAT\_ECV).

Type: uint8\_t

Default value: 2

**subcluster2.ete.CLAIMTAGS**

Number of claim tags.

Type: uint8\_t

Default value: 4

**subcluster2.ete.MAX\_INST\_PER\_Q**

Maximum limit for the number of instructions implied by a Q element.

Type: uint16\_t

Default value: 0x1

**subcluster2.ete.NumberOfRSPairs**

Number of resource selector pairs.

Type: uint8\_t

Default value: 8

**subcluster2.ete.PIDR\_CM0D**

TRCPIDR CM0D value.

Type: uint8\_t

Default value: 0

**subcluster2.ete.Q\_CADENCE**

Number of instruction blocks traced between two Q elements.

Type: `uint16_t`

Default value: `0x1`

#### **`subcluster2.ete.RES0_STATEFUL`**

Whether **RES0** bits are stateful or **RAZ/WI**.

Type: `bool`

Default value: `false`

#### **`subcluster2.ete.RETSTACK`**

Return stack depth.

Type: `uint8_t`

Default value: `3`

#### **`subcluster2.ete.SIM_OVERFLOW_GRANULARITY`**

Number of instruction blocks in each granule, for simulated overflow.

Type: `uint32_t`

Default value: `0x64`

#### **`subcluster2.ete.SIM_OVERFLOW_PERCENTAGE`**

Percentage of instruction blocks lost in each granule, for simulated overflow.

Type: `uint8_t`

Default value: `0`

#### **`subcluster2.ete.SOURCE_ADDRESS`**

Allow generation of source address elements.

Type: `bool`

Default value: `false`

#### **`subcluster2.ete.TRACE_OUTPUT`**

File to which to write trace byte stream.

Type: `string`

Default value: `N/A`

#### **`subcluster2.ete.TRCRSRTA_FORCED_EXCEP`**

TRCRSR.TA value for a forcibly traced exception.

Type: `bool`

Default value: `false`

### **`subcluster2.ext_abort_so_write_ras_type`**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `uint8_t`

Default value: 2

### **`subcluster2.force_mte_tag_access_razwi_and_ignore_tag_checks`**

Force MTE tag accesses to **RAZ/WI** and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

Type: `bool`

Default value: `false`

### **`subcluster2.force_zero_PSTATE_PAN`**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

### **`subcluster2.force_zero_mpam_partid_and_pmg`**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

Type: `uint8_t`

Default value: 0

### **`subcluster2.has_delayed_dbgreg`**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

### **`subcluster2.has_delayed_sysreg`**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

### **subcluster2.has\_enhanced\_pan**

Implements Armv8.7 Enhanced PAN feature (FEAT\_PAN3) values of this parameter are:- 1, feature is implemented if ARMv8.7 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 2

### **subcluster2.has\_ete**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (-plugin or -P).

Type: `bool`

Default value: `false`

### **subcluster2.has\_statistical\_profiling**

Whether Statistical Based Profiling is implemented (FEAT\_SPE).

Type: `bool`

Default value: `true`

### **subcluster2.has\_v8\_7\_spe\_inverted\_filtering**

Where FEAT\_SPEv1p2 is implemented, whether inverted filtering by events is implemented (represented by PMISDR.FnE).

Type: `bool`

Default value: `false`

### **subcluster2.has\_v8\_7\_spe\_previous\_branch\_target**

Where FEAT\_SPEv1p2 is implemented, whether the optional branch target feature is implemented (FEAT\_SPE\_PBT).

Type: `bool`

Default value: `false`

### **subcluster2.icache-hit\_latency**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **subcluster2.icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster2.icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster2.icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

Type: bool

Default value: false

### **subcluster2.icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster2.icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.icache-size**

L1 I-Cache size in bytes.

Type: uint32\_t

Default value: 0x10000

**subcluster2.instruction\_tlb\_size**

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

Type: uint32\_t

Default value: 0x0

**subcluster2.invalidate\_code\_cache\_on\_icache\_cmo**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: uint8\_t

Default value: 0

**subcluster2.memory\_tagging\_support\_level**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

Type: uint8\_t

Default value: 3

**subcluster2.mpam\_has\_altsp**

DEPRECATED: MPAMIDR\_EL1.HAS\_ALTSP is required when FEAT\_RME is implemented.

Type: bool

Default value: false

**subcluster2.mpamidr\_has\_force\_ns**

Whether MPAMIDR\_EL1.HAS\_FORCE\_NS bit is set or clear.values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.6 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

**subcluster2.mpamidr\_has\_sdeflt**

Whether MPAMIDR\_EL1.HAS\_SDEFLT bit is set or clear.values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.6 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

**subcluster2.mpamidr\_has\_tidr**

Whether MPAMIDR\_EL1.HAS\_TIDR bit is set or clear.values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.6 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

**subcluster2.pmu-num\_counters**

Number of PMU counters implemented.

Type: uint8\_t

Default value: 31

**subcluster2.ptw\_latency**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

**subcluster2.stage12\_tlb\_size**

Number of stage1+2 tlb entries. Valid values are 0 or >= 4.

Type: uint32\_t

Default value: 0x80

**subcluster2.tcr\_txsz\_undersize\_should\_fault**

If large VA is not supported, Whether undersized TxSZ value should generate translation fault.

Type: bool

Default value: false

**subcluster2.tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster2.tlbi_stall_enabled`**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: `bool`

Default value: `false`

#### **`subcluster2.treat_PAC_as_NOP`**

Non-architecture parameter. Treat Pointer Authentication as **NOP**. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are **NOP**. Brach, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

Type: `bool`

Default value: `false`

#### **`subcluster2.walk_cache_latency`**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

## 3.53 ARMCortexA520CT\_CortexA725CT

Defined in `LISA/ARMCortexA520CT_CortexA725CT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
CortexA520 r0p1	Full support
CortexA725 r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following parameters were added:

- `num_acp`
- `subcluster0.has_delayed_dbgreg`



- `subcluster0.has_delayed_sysreg`
- `subcluster1.has_delayed_dbgreg`
- `subcluster1.has_delayed_sysreg`

## About ARMCortexA520CT\_CortexA725CT

The number of cores in each subcluster is configurable using the following parameters:

### **`subcluster0.NUM_CORES`**

Possible values are 1-13 (ARMCortexA520CT).

### **`subcluster1.NUM_CORES`**

Possible values are 1-13 (ARMCortexA725CT).

The total number of cores in the cluster cannot exceed 14.

Port arrays are expanded to the width needed for the maximum number of cores in each subcluster. Use the following port array indexes:

- `<port_name>[0-12]` for cores in `subcluster0`.
- `<port_name>[13-25]` for cores in `subcluster1`.



All instances in the Master cross trigger matrix port array, `cti[26]` must be connected, regardless of the `NUM_CORES` value used.

Core-specific parameters have the following prefixes:

- `subcluster0.cpu0` to `subcluster0.cpu12` identify cores in `subcluster0`.
- `subcluster1.cpu0` to `subcluster1.cpu12` identify cores in `subcluster1`.

For information about the cores in this model, see:

- [ARMCortexA520CT](#).
- [ARMCortexA725CT](#).

## Iris and MTI instances for ARMCortexA520CT\_CortexA725CT

This model has the following Iris instances:

Name	Instance type
ARMCortexA520CT_CortexA725CT	Cluster_ARM_Cortex-A520_CortexA725_Heterogeneous
ARMCortexA520CT_CortexA725CT.AMU	PVBusLogger
ARMCortexA520CT_CortexA725CT.AMU.mapper	PVBusMapper
ARMCortexA520CT_CortexA725CT.DAP	PVBusLogger
ARMCortexA520CT_CortexA725CT.DAP.mapper	PVBusMapper
ARMCortexA520CT_CortexA725CT.DSU	DSU-120

Name	Instance type
ARMCortexA520CT_CortexA725CT.DSU.PPU_cluster	PPUv1
ARMCortexA520CT_CortexA725CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA520CT_CortexA725CT.DSU.PPU_coreZ (where Z = 0-1)	PPUv1
ARMCortexA520CT_CortexA725CT.DSU.PPU_coreZ.busslave (where Z = 0-1)	PVBusSlave
ARMCortexA520CT_CortexA725CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA520CT_CortexA725CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA520CT_CortexA725CT.DSU.shared_cache	PVCache
ARMCortexA520CT_CortexA725CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA520CT_CortexA725CT.DSU.shared_cache.upstream[Z] (where Z = 0-6)	PVBusSlave
ARMCortexA520CT_CortexA725CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA520CT_CortexA725CT.MMAP	PVBusLogger
ARMCortexA520CT_CortexA725CT.MMAP.mapper	PVBusMapper
ARMCortexA520CT_CortexA725CT.RAS	PVBusLogger
ARMCortexA520CT_CortexA725CT.RAS.mapper	PVBusMapper
ARMCortexA520CT_CortexA725CT.cpuZ.debug_rom (where Z = 0-1)	debug_rom
ARMCortexA520CT_CortexA725CT.ext_bus	PVBusLogger
ARMCortexA520CT_CortexA725CT.ext_bus.mapper	PVBusMapper
ARMCortexA520CT_CortexA725CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA520CT_CortexA725CT.global_debug_rom	debug_rom
ARMCortexA520CT_CortexA725CT.secondary_debug_rom	debug_rom
ARMCortexA520CT_CortexA725CT.subcluster0	Subcluster_ARM_Cortex-A520
ARMCortexA520CT_CortexA725CT.subcluster0.cpu0	ARM_Cortex-A520
ARMCortexA520CT_CortexA725CT.subclusterZ.cpu0.UTLB (where Z = 0-1)	TLB
ARMCortexA520CT_CortexA725CT.subclusterZ.cpuU.dtlb (where Z = 0-1; U = 0-1)	TLB
ARMCortexA520CT_CortexA725CT.subclusterZ.cpu0.l1dcache (where Z = 0-1)	PVCache
ARMCortexA520CT_CortexA725CT.subclusterZ.cpu0.l1dcache.downstream[0].pvbusmaster (where Z = 0-1)	PVBusMaster
ARMCortexA520CT_CortexA725CT.subclusterZ.cpu0.l1dcache.upstream[0] (where Z = 0-1)	PVBusSlave
ARMCortexA520CT_CortexA725CT.subclusterZ.cpu0.l1icache (where Z = 0-1)	PVCache
ARMCortexA520CT_CortexA725CT.subclusterZ.cpu0.l1icache.downstream[0].pvbusmaster (where Z = 0-1)	PVBusMaster
ARMCortexA520CT_CortexA725CT.subclusterZ.cpu0.l1icache.upstream[0] (where Z = 0-1)	PVBusSlave
ARMCortexA520CT_CortexA725CT.subclusterZ.cpu0.l2cache (where Z = 0-1)	PVCache
ARMCortexA520CT_CortexA725CT.subclusterZ.cpu0.l2cache.downstream[0].pvbusmaster (where Z = 0-1)	PVBusMaster
ARMCortexA520CT_CortexA725CT.subclusterZ.cpu0.l2cache.upstream[W] (where Z = 0-1; W = 0-1)	PVBusSlave
ARMCortexA520CT_CortexA725CT.subclusterZ.sve (where Z = 0-1)	ScalableVectorExtension

Name	Instance type
ARMCortexA520CT_CortexA725CT.subcluster1	Subcluster_ARM_Cortex-A725
ARMCortexA520CT_CortexA725CT.subcluster1.cpu0	ARM_Cortex-A725

This model has the following MTI trace components:

Name	Component type
ARMCortexA520CT_CortexA725CT.AMU	PVBusLogger
ARMCortexA520CT_CortexA725CT.AMU.mapper	PVBusMapper
ARMCortexA520CT_CortexA725CT.DAP	PVBusLogger
ARMCortexA520CT_CortexA725CT.DAP.mapper	PVBusMapper
ARMCortexA520CT_CortexA725CT.DSU	DSU-120
ARMCortexA520CT_CortexA725CT.DSU.PPU_cluster	PPUv1
ARMCortexA520CT_CortexA725CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA520CT_CortexA725CT.DSU.PPU_coreZ (where Z = 0-1)	PPUv1
ARMCortexA520CT_CortexA725CT.DSU.PPU_coreZ.busslave (where Z = 0-1)	PVBusSlave
ARMCortexA520CT_CortexA725CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA520CT_CortexA725CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA520CT_CortexA725CT.DSU.shared_cache	PVCache
ARMCortexA520CT_CortexA725CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA520CT_CortexA725CT.DSU.shared_cache.upstream[Z] (where Z = 0-6)	PVBusSlave
ARMCortexA520CT_CortexA725CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA520CT_CortexA725CT.MMAP	PVBusLogger
ARMCortexA520CT_CortexA725CT.MMAP.mapper	PVBusMapper
ARMCortexA520CT_CortexA725CT.RAS	PVBusLogger
ARMCortexA520CT_CortexA725CT.RAS.mapper	PVBusMapper
ARMCortexA520CT_CortexA725CT.ext_bus	PVBusLogger
ARMCortexA520CT_CortexA725CT.ext_bus.mapper	PVBusMapper
ARMCortexA520CT_CortexA725CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA520CT_CortexA725CT.subcluster0.cpu0	ARM_Cortex-A520
ARMCortexA520CT_CortexA725CT.subclusterZ.cpu0.UTLB (where Z = 0-1)	TLB
ARMCortexA520CT_CortexA725CT.subclusterZ.cpu0.l1dcache (where Z = 0-1)	PVCache
ARMCortexA520CT_CortexA725CT.subclusterZ.cpu0.l1dcache.downstream[0].pvbusmaster (where Z = 0-1)	PVBusMaster
ARMCortexA520CT_CortexA725CT.subclusterZ.cpu0.l1dcache.upstream[0] (where Z = 0-1)	PVBusSlave
ARMCortexA520CT_CortexA725CT.subclusterZ.cpu0.l1icache (where Z = 0-1)	PVCache
ARMCortexA520CT_CortexA725CT.subclusterZ.cpu0.l1icache.downstream[0].pvbusmaster (where Z = 0-1)	PVBusMaster
ARMCortexA520CT_CortexA725CT.subclusterZ.cpu0.l1icache.upstream[0] (where Z = 0-1)	PVBusSlave
ARMCortexA520CT_CortexA725CT.subclusterZ.cpu0.l2cache (where Z = 0-1)	PVCache

Name	Component type
ARMCortexA520CT_CortexA725CT.subclusterZ.cpu0.l2cache.downstream[0].pvbusmaster (where Z = 0-1)	PVBusMaster
ARMCortexA520CT_CortexA725CT.subclusterZ.cpu0.l2cache.upstream[W] (where Z = 0-1; W = 0-1)	PVBusSlave
ARMCortexA520CT_CortexA725CT.subcluster1.cpu0	ARM_Cortex-A725

### Ports for ARMCortexA520CT\_CortexA725CT

Port	Direction	Protocol	Description
acp_s	slave	PVBus	AXI ACP slave port.
AEND0MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND2MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND3MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART2MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	slave	Signal	CHI defined pins.
broadcastcachemaint	slave	Signal	ACE defined pins.
broadcastouter	slave	Signal	ACE defined pins.
broadcastpersist	slave	Signal	CHI defined pins.
cfgend	slave	Signal	This signal is for EE bit initialisation.
cfgsdisable	slave	Signal	This signal disables write access to some secure Interrupt Controller registers.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcs_m_pchannel	master	PChannel	Cluster PCSM signal
cluster_powerdown_out	master	Signal	DEPRECATED - An output from PPUs that informs thermal controller of the cluster power info
cluster_ppu_hw_stat	master	Value	Cluster PPU power state

Port	Direction	Protocol	Description
clusterid	slave	Value	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set ManagerID64 of the core.
clusterpmuirq	master	Signal	DynamiQ pmu irq
CNTHPIRQ	master	Signal	Timer signals to SOC.
CNTHPSIRQ	master	Signal	Timer signals to SOC
CNTHVIRQ	master	Signal	Timer signals to SOC.
CNTHVSIQ	master	Signal	Timer signals to SOC
CNTPNSIRQ	master	Signal	Timer signals to SOC.
CNTPSIQ	master	Signal	Timer signals to SOC.
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.
CNTVIRQ	master	Signal	Timer signals to SOC.
commirq	master	Signal	Interrupt signal from debug communications channel.
core_clk_in	slave	ClockSignal	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
core_pcsn_pchannel	master	PChannel	Core PCSM signals
core_powerdown_out	master	Signal	DEPRECATED - An output from PPUs that informs thermal controller of the core power info
core_ppu_hw_stat	master	Value	Notify the power state change inside the PPU.
coreerrirq	master	Signal	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq	master	Signal	Fault indicator for a detected 1 or 2 bit ECC error
coreinstrrun	master	Signal	Core Running State
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
ctidbgirq	master	Signal	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	slave	Signal	External debug interface.
dbgnopwrdown	master	Signal	No power-down request.
dbgprupreq	master	Signal	Debug power up request.
defaultMP	slave	Signal	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main master interface ports.
dev_debug_s	slave	PVBus	External debug interface.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.

Port	Direction	Protocol	Description
gicreset	master	Signal	An output from PPU that can be used to reset GICCLK domain components
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	slave	PVBus	External debug interface.
pmbirq	master	Signal	Interrupt signal from the statistical profiling unit.
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	master	Signal	PPU cluster interrupt.
ppu_cluster_isolate	master	Signal	PPU Cluster Domain isolation control.
ppu_cluster_wakerequest	slave	Signal	PPU cluster wake request signal.
ppu_core_irq	master	Signal	PPU core interrupt.
ppu_core_isolate	master	Signal	PPU Core Domain isolation control.
ppu_core_wakerequest	slave	Signal	PPU core wake request signal.
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	master	PVBus	The core will generate bus requests on this port.
pvbus_periph_m	master	PVBus	The core can generate peripheral bus request on this port.
rei	slave	Signal	Per core RAM Error Interrupt.
reset	slave	Signal	Reset signal to cluster.
rvbaraddr	slave	Value_64	Reset vector base address.
sci_m	master	SystemCoherencyInterface	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei	slave	Signal	Per core System Error physical pins.
spiden	slave	Signal	External debug interface.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq	master	Signal	Interrupt signal from the trace buffer unit.
utility_bus_s	slave	PVBus	Utility bus slave
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfiq	slave	Signal	Virtualised FIQ.
virq	slave	Signal	Virtualised IRQ.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Per core virtual System Error physical pins.

## Parameters for ARMCortexA520CT\_CortexA725CT

### **AEND0\_DEFAULT**

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

Type: `uint64_t`

Default value: `0x0`

### **AEND1\_DEFAULT**

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

Type: `uint64_t`

Default value: `0x0`

### **AEND2\_DEFAULT**

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

Type: `uint64_t`

Default value: `0x0`

### **AEND3\_DEFAULT**

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

Type: `uint64_t`

Default value: `0x0`

### **ASTART0\_DEFAULT**

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

Type: `uint64_t`

Default value: `0x0`

### **ASTART1\_DEFAULT**

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

Type: `uint64_t`

Default value: 0x0

### **ASTART2\_DEFAULT**

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

Type: `uint64_t`

Default value: 0x0

### **ASTART3\_DEFAULT**

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

Type: `uint64_t`

Default value: 0x0

### **BROADCASTATOMIC**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`

Default value: `true`

### **BROADCASTCACHEMAINT**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

Type: `bool`

Default value: `false`

### **BROADCASTOUTER**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

Type: `bool`

Default value: `false`

### **BROADCASTPERSIST**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: `bool`



Default value: `true`

### **CLUSTER\_ID**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the ManagerID64 of the core.

Type: `uint32_t`

Default value: `0x0`

### **CMO\_broadcast\_when\_cache\_state\_modelling\_disabled**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. This value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain events even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required.

Type: `uint8_t`

Default value: `1`

### **DBGROMADDR**

Initialization value of DBGDRAR register. Bits[55:12] of this register specify the ROM table physical address.

Type: `uint64_t`

Default value: `0x0`

### **DBGROMADDRV**

If true, set bits[1:0] of the CP15 DBGDRAR to indicate that the address is valid.

Type: `bool`

Default value: `false`

### **GICDISABLE**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`

Default value: `true`

**bus\_type**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

Type: `uint8_t`

Default value: 0

**core\_complex\_mapping**

Defines Complex descriptions for platforms that support several Cores per Complex like Cortex-A510. JSON format: {"complex0": { "cores" : [ 0, 1 ], "l2-cache" : {"exists":1, "size":16MB}}, ... , "complexN": { "cores" : [<core\_list>, "l2-cache" : {"exists":1, "size":16MB}}} where <core\_list> is the list of cores in the complexN. Effective only when the parameter value is not empty.

Type: `string`

Default value: {"complex0": { "cores": [0, 1], "l2-cache" : {"exists":1, "size": "16MB"}}, "complex1": { "cores": [2, 3], "l2-cache" : {"exists":1, "size": "16MB"}}, "complex2": { "cores": [4, 5], "l2-cache" : {"exists":1, "size": "16MB"}}, "complex3": { "cores": [6, 7], "l2-cache" : {"exists":1, "size": "16MB"}}, "complex4": { "cores": [8, 9], "l2-cache" : {"exists":1, "size": "16MB"}}, "complex5": { "cores": [10, 11], "l2-cache" : {"exists":1, "size": "16MB"}}, "complex6": { "cores": [12, 13], "l2-cache" : {"exists":1, "size": "16MB"}} }

**core\_power\_on\_by\_default**

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPUs which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

Type: `bool`

Default value: `false`

**dcache-state\_modelled**

Set whether D-cache has stateful implementation.

Type: `bool`

Default value: `false`

**enable\_simulation\_performance\_optimizations**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of `stage12_tlb_size` parameter to 1024).

Type: `bool`

Default value: `true`

### **has\_peripheral\_port**

If true, an additional AXI peripheral port is configured.

Type: `bool`

Default value: `false`

### **icache-state\_modelled**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **l3cache-has\_mpam**

L3 Cache has MPAM support.

Type: `bool`

Default value: `true`

### **l3cache-hit\_latency**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **l3cache-maintenance\_latency**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **l3cache-miss\_latency**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-read\_access\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size l3cache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**l3cache-read\_bus\_width\_in\_bytes**

L3 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**l3cache-read\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. l3cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**l3cache-size**

L3 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**l3cache-snoop\_data\_transfer\_latency**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**l3cache-snoop\_issue\_latency**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **l3cache-ways**

L3 Cache number of ways (sets are implicit from size).

Type: `uint16_t`

Default value: 16

### **l3cache-write\_access\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size `l3cache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **l3cache-write\_bus\_width\_in\_bytes**

L3 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: `uint32_t`

Default value: 0x10

### **l3cache-write\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l3cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **mpmm\_accumulator\_multiplier**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator. value of `n` means the accumulator will use (`n * accumulator value`) to calculate the mpmm threshold (MPMM). is provided as a fast model workaround to handle cases where execution of quantums in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

Type: `uint8_t`

Default value: 1

### **num\_acp**

Number of ACP ports.

Type: `uint8_t`

Default value: 0

### **num\_nodes**

Number of transport nodes. Zero implies direct-connect configuration.

Type: `uint8_t`

Default value: 1

### **revision\_number**

Revision number of TRM implemented by the model. Changing the value can change the model behaviour. Visible as the Variant field in MIDR/MIDR\_EL1. Corresponds to the revision number X in rXpY.

Type: `uint8_t`

Default value: 0

### **subcluster0.CPUCFR**

Value of CPU Configuration Register.

Type: `uint64_t`

Default value: 0x0

### **subcluster0.NUM\_CORES**

Number of cores in the subcluster. Total number of cores in cluster may not exceed 14.

Type: `uint8_t`

Default value: 1

### **subcluster0.core\_cache\_protection**

core\_cache\_protection can change ERROFR, ERROPFGF and ERROPFGCTL fields. Possible values are: -1:Not implemented (by default), 0:Disabled, 1:Enabled.

Type: `int8_t`

Default value: 1

### **subcluster0.cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: 0x1

**subcluster0.cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**subcluster0.cpu0.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster0.cpu0.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster0.cpu0.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu0.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu0.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu0.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu0.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu0.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu0.l2cache-read_bus_width_in_bytes`**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: `uint32_t`

Default value: `0x10`

### **`subcluster0.cpu0.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu0.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`



**subcluster0.cpu0.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu0.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu0.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

**subcluster0.cpu0.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

**subcluster0.cpu0.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu0.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu0.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu0.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster0.cpu0.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster0.cpu0.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu0.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu0.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu0.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu0.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu0.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu0.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: 0xf000000

**subcluster0.cpu0.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu0.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster0.cpu0.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: true

**subcluster0.cpu1.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: false

**subcluster0.cpu1.CRYPTODISABLE**

Disable cryptographic features.

Type: bool

Default value: false

**subcluster0.cpu1.RVBARADDR**

Value of RVBAR\_ELx register.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter trace\_special\_hlt\_imm16.

Type: bool

Default value: false

**subcluster0.cpu1.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu1.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu1.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu1.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu1.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu1.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu1.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu1.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu1.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**`subcluster0.cpu1.semihosting-T32_HLT`**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**`subcluster0.cpu1.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**`subcluster0.cpu1.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**`subcluster0.cpu1.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**`subcluster0.cpu1.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**`subcluster0.cpu1.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**`subcluster0.cpu1.semihosting-heap_limit`**

Virtual address of top of heap.



Type: `uint64_t`

Default value: `0xf0000000`

### **`subcluster0.cpu1.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster0.cpu1.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf0000000`

### **`subcluster0.cpu1.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu1.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu1.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu10.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster0.cpu10.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster0.cpu10.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu10.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu10.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu10.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu10.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu10.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu10.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu10.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu10.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu10.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu10.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu10.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

#### **subcluster0.cpu10.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu10.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

#### **subcluster0.cpu10.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu10.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

#### **subcluster0.cpu10.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu10.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu10.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu10.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu10.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster0.cpu10.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster0.cpu10.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu10.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu10.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: true

**subcluster0.cpu10.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu10.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu10.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster0.cpu10.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu10.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu10.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu10.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu11.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu11.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu11.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu11.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu11.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu11.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu11.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu11.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu11.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: `uint32_t`

Default value: `0x10`

**subcluster0.cpu11.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu11.l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`



Default value: 0x80000

**subcluster0.cpu11.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu11.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu11.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu11.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu11.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu11.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu11.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu11.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu11.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu11.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu11.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu11.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster0.cpu11.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

### **`subcluster0.cpu11.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **`subcluster0.cpu11.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **`subcluster0.cpu11.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster0.cpu11.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu11.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu11.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster0.cpu11.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

#### **`subcluster0.cpu11.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster0.cpu11.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu11.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster0.cpu12.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu12.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu12.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: 0x0

**subcluster0.cpu12.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter trace\_special\_hlt\_imm16.

Type: bool

Default value: false

**subcluster0.cpu12.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu12.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu12.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu12.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu12.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

#### **subcluster0.cpu12.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu12.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

#### **subcluster0.cpu12.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu12.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu12.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

#### **subcluster0.cpu12.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu12.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu12.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu12.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu12.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu12.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu12.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu12.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu12.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster0.cpu12.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster0.cpu12.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu12.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu12.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu12.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`



**subcluster0.cpu12.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu12.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu12.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu12.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu12.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster0.cpu12.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster0.cpu2.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster0.cpu2.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster0.cpu2.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu2.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu2.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu2.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu2.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu2.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu2.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu2.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu2.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu2.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu2.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu2.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu2.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu2.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster0.cpu2.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster0.cpu2.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster0.cpu2.semihosting-cwd**

Base directory for semihosting file access.

Type: string

Default value: N/A

**subcluster0.cpu2.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: true

**subcluster0.cpu2.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu2.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster0.cpu2.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu2.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu2.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu2.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu3.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu3.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu3.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu3.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu3.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu3.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t



Default value: 0x0

#### **subcluster0.cpu3.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

#### **subcluster0.cpu3.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu3.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu3.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

#### **subcluster0.cpu3.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu3.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu3.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu3.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu3.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu3.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu3.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu3.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

### **`subcluster0.cpu3.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

### **`subcluster0.cpu3.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **`subcluster0.cpu3.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **`subcluster0.cpu3.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster0.cpu3.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu3.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu3.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster0.cpu3.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu3.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu3.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu3.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu4.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu4.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu4.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu4.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu4.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu4.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu4.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu4.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster0.cpu4.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu4.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu4.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu4.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu4.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu4.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu4.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu4.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu4.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu4.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster0.cpu4.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster0.cpu4.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu4.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu4.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`



**subcluster0.cpu4.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu4.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu4.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu4.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu4.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu4.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster0.cpu4.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu5.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu5.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu5.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu5.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu5.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu5.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu5.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu5.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu5.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu5.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu5.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu5.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu5.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu5.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu5.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu5.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster0.cpu5.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster0.cpu5.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster0.cpu5.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster0.cpu5.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu5.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu5.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu5.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu5.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu5.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster0.cpu5.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu5.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster0.cpu6.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu6.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu6.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu6.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu6.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu6.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t



Default value: 0x0

**subcluster0.cpu6.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu6.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu6.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu6.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu6.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu6.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu6.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu6.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu6.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

### **`subcluster0.cpu6.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

### **`subcluster0.cpu6.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **`subcluster0.cpu6.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **`subcluster0.cpu6.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster0.cpu6.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu6.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu6.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster0.cpu6.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu6.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu6.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu6.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu7.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu7.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster0.cpu7.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu7.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu7.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu7.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu7.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu7.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster0.cpu7.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu7.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu7.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu7.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu7.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu7.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu7.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu7.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu7.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu7.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster0.cpu7.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster0.cpu7.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu7.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu7.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`



**subcluster0.cpu7.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu7.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster0.cpu7.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu7.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu7.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster0.cpu7.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu8.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu8.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu8.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu8.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu8.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu8.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu8.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu8.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu8.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu8.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu8.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu8.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu8.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu8.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu8.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu8.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster0.cpu8.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster0.cpu8.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster0.cpu8.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster0.cpu8.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu8.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu8.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu8.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu8.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu8.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster0.cpu8.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu8.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster0.cpu9.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu9.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu9.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu9.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu9.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu9.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t



Default value: 0x0

#### **subcluster0.cpu9.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

#### **subcluster0.cpu9.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu9.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu9.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

#### **subcluster0.cpu9.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu9.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu9.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu9.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu9.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu9.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu9.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu9.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster0.cpu9.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster0.cpu9.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster0.cpu9.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster0.cpu9.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu9.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: 0x0

**subcluster0.cpu9.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: 0xf000000

**subcluster0.cpu9.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster0.cpu9.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu9.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu9.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu9.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.dcache-hit_latency`**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.dcache-maintenance_latency`**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

#### **subcluster0.dcache-miss\_latency**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.dcache-prefetch\_enabled**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

Type: bool

Default value: false

#### **subcluster0.dcache-read\_access\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.dcache-size**

L1 D-Cache size in bytes.

Type: uint32\_t

Default value: 0x8000

#### **subcluster0.dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.dcache-write_access_latency`**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.dcache-write_latency`**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.ete.CLAIMTAGS`**

Number of claim tags.

Type: `uint8_t`

Default value: 4

#### **`subcluster0.ete.MAX_INST_PER_Q`**

Maximum limit for the number of instructions implied by a Q element.

Type: `uint16_t`

Default value: `0x1`

#### **`subcluster0.ete.PIDR_CM0D`**

TRCPIDR CM0D value.

Type: `uint8_t`

Default value: 0

#### **`subcluster0.ete.PIDR_REVAND`**

TRCPIDR REVAND value.

Type: `uint8_t`

Default value: 0

**subcluster0.ete.PIDR\_REVISION**

TRCPIDR REVISION value.

Type: uint8\_t

Default value: 0

**subcluster0.ete.Q\_CADENCE**

Number of instruction blocks traced between two Q elements.

Type: uint16\_t

Default value: 0x1

**subcluster0.ete.RES0\_STATEFUL**

Whether **RES0** bits are stateful or **RAZ/WI**.

Type: bool

Default value: false

**subcluster0.ete.RETSTACK**

Return stack depth.

Type: uint8\_t

Default value: 3

**subcluster0.ete.REVISION**

TRCIDR1 revision value.

Type: uint8\_t

Default value: 0

**subcluster0.ete.SIM\_OVERFLOW\_GRANULARITY**

Number of instruction blocks in each granule, for simulated overflow.

Type: uint32\_t

Default value: 0x64

**subcluster0.ete.SIM\_OVERFLOW\_PERCENTAGE**

Percentage of instruction blocks lost in each granule, for simulated overflow.

Type: uint8\_t

Default value: 0

**subcluster0.ete.SOURCE\_ADDRESS**

Allow generation of source address elements.

Type: `bool`

Default value: `false`

**subcluster0.ete.TRACE\_OUTPUT**

File to which to write trace byte stream.

Type: `string`

Default value: `N/A`

**subcluster0.ete.TRCRSRTA\_FORCED\_EXCEP**

TRCRSR.TA value for a forcibly traced exception.

Type: `bool`

Default value: `false`

**subcluster0.force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks**

Force MTE tag accesses to **RAZ/WI** and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

Type: `bool`

Default value: `false`

**subcluster0.force\_zero\_PSTATE\_PAN**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

**subcluster0.force\_zero\_mpam\_partid\_and\_pmg**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

Type: `uint8_t`



Default value: 0

**subcluster0.has\_actlr2**

If true ACLTR2 exists and ACTLR2(NS) is aliased to ACTLR\_EL1[63:32].

Type: bool

Default value: true

**subcluster0.has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: bool

Default value: false

**subcluster0.has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: bool

Default value: true

**subcluster0.has\_dot\_product**

Possible values of this parameter are:- 1, feature not implemented.- 2, feature is implemented.

Type: uint8\_t

Default value: 2

**subcluster0.has\_ete**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (-plugin or -P).

Type: bool

Default value: false

**subcluster0.icache-hit\_latency**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

**subcluster0.icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.icache-size**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: `0x8000`

#### **`subcluster0.invalidate_code_cache_on_icache_cmo`**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: `uint8_t`

Default value: `0`

#### **`subcluster0.memory_tagging_support_level`**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

Type: `uint8_t`

Default value: `3`

#### **`subcluster0.pmu-num_counters`**

Number of PMU counters implemented.

Type: `uint8_t`

Default value: `6`

#### **`subcluster0.ptw_latency`**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.ras_extra_configurations`**

Miscellaneous configurations for error records. An array of JSON objects. Note for `ERXCTLR_EL1` register it only allows to define the mask value for the `IMPDEF` fields, ie bits [63:32] and bit 1, but its reset value applies on all fields. Note for `ERXMISCN` masks - these are 64 bit masks covering the 64 bit registers `ERXMISCN_EL1`. E.g. [{"Index": 0, "ERXMISCO\_mask": 0x0, "ERXMISCO\_reset": 0x0, "ERXMISC1\_mask": 0x0, "ERXMISC1\_reset": 0x0, "ERXMISC2\_mask": 0x0, "ERXMISC2\_reset": 0x0, "ERXMISC3\_mask": 0x0, "ERXMISC3\_reset": 0x0, "ERXCTLR\_EL1\_mask": 0x0, "ERXCTLR\_EL1\_reset": 0x0}, {"Index": 1, "ERXMISCO\_mask": 0x0, "ERXMISCO\_reset": 0x0, "ERXSTATUS\_IERR\_mask": 0x300}].

Type: `string`

Default value: "[ { \"Index\": 1, \"ERXMISC0\_mask\": 0xFFFFc0003fc3, \"ERXMISC1\_mask\": 0x03F870003FF30f07, \"ERXPGCTL\_reset\": 0x1000 }, { \"Index\": 2, \"ERXMISC0\_mask\": 0xFFFFe007ffc0, \"ERXMISC0\_reset\": 0x2, \"ERXSTATUS\_IERR\_mask\": 0x300 , \"ERXMISC1\_mask\": 0xFF8700ffff31f0f, \"ERXPGCTL\_reset\": 0x1000 } ]"

### **subcluster0.ras\_pfg\_clock\_mhz**

RAS Pseudo-Fault generation clock rate in MHz.

Type: uint8\_t

Default value: 12

### **subcluster0.revision\_number**

Revision number of TRM implemented by the model. Changing the value can change the model behaviour. Visible as the Variant field in MIDR/MIDR\_EL1. Corresponds to the revision number X in rXpY.

Type: uint8\_t

Default value: 0

### **subcluster0.tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

### **subcluster0.tlbi\_stall\_enabled**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: bool

Default value: false

### **subcluster0.treat\_PAC\_as\_NOP**

Non-architecture parameter. Treat Pointer Authentication as **NOP**. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are **NOP**. Branch, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

Type: bool

Default value: false

### **subcluster0.walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.CPUCFR`**

Value of CPU Configuration Register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.NUM_CORES`**

Number of cores in the subcluster. Total number of cores in cluster may not exceed 14.

Type: `uint8_t`

Default value: `1`

### **`subcluster1.cpi_div`**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

### **`subcluster1.cpi_mul`**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

### **`subcluster1.cpu0.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster1.cpu0.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster1.cpu0.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu0.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu0.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu0.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu0.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu0.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu0.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu0.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu0.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu0.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x100000`

#### **`subcluster1.cpu0.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu0.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu0.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu0.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu0.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu0.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu0.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu0.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000



**subcluster1.cpu0.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu0.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster1.cpu0.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster1.cpu0.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu0.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu0.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu0.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu0.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu0.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu0.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu0.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu0.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu0.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster1.cpu1.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu1.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster1.cpu1.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu1.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu1.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu1.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu1.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x100000`

**subcluster1.cpu1.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu1.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu1.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu1.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu1.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu1.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu1.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu1.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu1.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu1.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster1.cpu1.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster1.cpu1.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu1.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu1.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu1.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu1.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu1.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu1.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu1.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu1.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu1.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster1.cpu10.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster1.cpu10.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu10.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu10.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu10.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`



**subcluster1.cpu10.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu10.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu10.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu10.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu10.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu10.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

#### **subcluster1.cpu10.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x100000

#### **subcluster1.cpu10.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu10.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu10.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu10.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu10.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu10.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu10.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu10.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu10.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu10.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster1.cpu10.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster1.cpu10.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster1.cpu10.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster1.cpu10.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu10.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu10.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu10.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu10.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu10.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu10.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu10.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster1.cpu11.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster1.cpu11.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu11.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu11.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu11.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter trace\_special\_hlt\_imm16.

Type: `bool`

Default value: `false`

**subcluster1.cpu11.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu11.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu11.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu11.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu11.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu11.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu11.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x100000

**subcluster1.cpu11.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu11.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu11.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu11.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu11.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu11.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu11.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu11.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu11.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu11.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster1.cpu11.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.



Type: `uint8_t`

Default value: 171

**`subcluster1.cpu1.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**`subcluster1.cpu1.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**`subcluster1.cpu1.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**`subcluster1.cpu1.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**`subcluster1.cpu1.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**`subcluster1.cpu1.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**`subcluster1.cpu1.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

#### **`subcluster1.cpu11.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster1.cpu11.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu11.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster1.cpu12.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu12.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu12.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu12.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu12.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu12.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu12.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu12.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu12.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu12.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu12.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu12.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x100000

**subcluster1.cpu12.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu12.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu12.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu12.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu12.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

### **`subcluster1.cpu12.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

### **`subcluster1.cpu12.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu12.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu12.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu12.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster1.cpu12.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster1.cpu12.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster1.cpu12.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster1.cpu12.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu12.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: 0x0

**subcluster1.cpu12.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: 0xf000000

**subcluster1.cpu12.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu12.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu12.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu12.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu12.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster1.cpu2.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu2.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster1.cpu2.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu2.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu2.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu2.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.



Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu2.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu2.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu2.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu2.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x100000`

### **`subcluster1.cpu2.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu2.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu2.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu2.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu2.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu2.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu2.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu2.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu2.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster1.cpu2.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster1.cpu2.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu2.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu2.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu2.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu2.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu2.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster1.cpu2.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu2.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu2.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster1.cpu2.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster1.cpu3.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster1.cpu3.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu3.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu3.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu3.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu3.force-fpsid`**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu3.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x100000

**subcluster1.cpu3.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu3.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu3.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu3.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu3.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu3.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster1.cpu3.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster1.cpu3.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster1.cpu3.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`



Default value: `N/A`

**subcluster1.cpu3.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu3.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu3.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu3.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu3.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu3.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu3.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster1.cpu4.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu4.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster1.cpu4.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu4.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu4.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu4.force-fpsid`**

Override the FPSID value.

Type: `bool`

Default value: `true`

### **`subcluster1.cpu4.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu4.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu4.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu4.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu4.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to

the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu4.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x100000`

#### **`subcluster1.cpu4.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu4.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu4.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu4.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu4.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu4.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu4.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu4.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu4.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu4.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster1.cpu4.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster1.cpu4.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu4.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu4.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu4.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu4.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu4.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu4.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: 0xf000000

**subcluster1.cpu4.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu4.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster1.cpu4.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: true

**subcluster1.cpu5.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: false

**subcluster1.cpu5.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: bool

Default value: false

**subcluster1.cpu5.CRYPTODISABLE**

Disable cryptographic features.

Type: bool

Default value: false

**subcluster1.cpu5.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu5.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu5.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu5.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu5.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu5.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`



**subcluster1.cpu5.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu5.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu5.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x100000

**subcluster1.cpu5.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu5.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu5.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu5.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu5.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

### **`subcluster1.cpu5.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

### **`subcluster1.cpu5.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu5.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu5.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu5.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster1.cpu5.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster1.cpu5.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster1.cpu5.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster1.cpu5.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu5.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu5.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu5.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu5.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu5.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu5.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu5.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster1.cpu6.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu6.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster1.cpu6.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu6.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu6.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu6.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu6.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu6.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu6.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu6.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu6.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu6.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x100000`

### **`subcluster1.cpu6.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu6.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu6.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu6.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu6.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu6.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu6.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu6.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu6.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu6.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster1.cpu6.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster1.cpu6.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu6.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu6.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`



**subcluster1.cpu6.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu6.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu6.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster1.cpu6.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu6.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu6.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster1.cpu6.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster1.cpu7.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu7.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu7.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu7.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu7.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu7.force-fpsid`**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu7.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu7.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu7.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu7.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu7.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu7.l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: 0x100000

#### **subcluster1.cpu7.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu7.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu7.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu7.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu7.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

#### **subcluster1.cpu7.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu7.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster1.cpu7.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster1.cpu7.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: 0x123456

**subcluster1.cpu7.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster1.cpu7.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster1.cpu7.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster1.cpu7.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster1.cpu7.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: true

**subcluster1.cpu7.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu7.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu7.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster1.cpu7.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu7.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu7.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu7.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster1.cpu8.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu8.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster1.cpu8.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu8.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu8.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu8.force-fpsid`**

Override the FPSID value.

Type: `bool`

Default value: `true`

### **`subcluster1.cpu8.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu8.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu8.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu8.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu8.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to



the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu8.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x100000`

#### **`subcluster1.cpu8.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu8.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu8.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu8.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu8.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu8.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu8.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu8.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu8.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu8.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster1.cpu8.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster1.cpu8.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu8.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu8.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu8.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu8.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu8.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu8.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: 0xf000000

**subcluster1.cpu8.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu8.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster1.cpu8.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: true

**subcluster1.cpu9.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: false

**subcluster1.cpu9.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: bool

Default value: false

**subcluster1.cpu9.CRYPTODISABLE**

Disable cryptographic features.

Type: bool

Default value: false

**subcluster1.cpu9.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu9.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu9.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu9.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu9.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu9.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu9.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x100000

**subcluster1.cpu9.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu9.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu9.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

### **`subcluster1.cpu9.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

### **`subcluster1.cpu9.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu9.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu9.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu9.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster1.cpu9.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster1.cpu9.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster1.cpu9.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster1.cpu9.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu9.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: 0x0

**subcluster1.cpu9.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: 0xf000000



**subcluster1.cpu9.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu9.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu9.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu9.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu9.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster1.dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.dcache-maintenance\_latency**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.dcache-miss\_latency**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.dcache-prefetch\_enabled**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

Type: bool

Default value: false

**subcluster1.dcache-read\_access\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.dcache-size**

L1 D-Cache size in bytes.

Type: uint32\_t

Default value: 0x10000

**subcluster1.dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.dcache-write\_access\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.ete.CLAIMTAGS**

Number of claim tags.

Type: uint8\_t

Default value: 4

**subcluster1.ete.ETE\_REVISION**

ETE revision: 0=ETEv1.0, 1=ETEv1.1, 2=ETEv1.2, 3=ETEv1.3.

Type: uint8\_t

Default value: 1

**subcluster1.ete.MAX\_INST\_PER\_Q**

Maximum limit for the number of instructions implied by a Q element.

Type: uint16\_t

Default value: 0x1

**subcluster1.ete.NumberOfRSPairs**

Number of resource selector pairs.

Type: uint8\_t

Default value: 8

**subcluster1.ete.PIDR\_CM0D**

TRCPIDR CM0D value.

Type: uint8\_t

Default value: 0

**subcluster1.ete.PIDR\_REVAND**

TRCPIDR REVAND value.

Type: uint8\_t

Default value: 0

**subcluster1.ete.PIDR\_REVISION**

TRCPIDR REVISION value.

Type: uint8\_t

Default value: 0

**subcluster1.ete.Q\_CADENCE**

Number of instruction blocks traced between two Q elements.

Type: uint16\_t

Default value: 0x1

**subcluster1.ete.RES0\_STATEFUL**

Whether RES0 bits are stateful or RAZ/WI.

Type: bool

Default value: false

**subcluster1.ete.RETSTACK**

Return stack depth.

Type: uint8\_t

Default value: 1

**subcluster1.ete.REVISION**

TRCIDR1 revision value.

Type: `uint8_t`

Default value: 0

**subcluster1.ete.SIM\_OVERFLOW\_GRANULARITY**

Number of instruction blocks in each granule, for simulated overflow.

Type: `uint32_t`

Default value: 0x64

**subcluster1.ete.SIM\_OVERFLOW\_PERCENTAGE**

Percentage of instruction blocks lost in each granule, for simulated overflow.

Type: `uint8_t`

Default value: 0

**subcluster1.ete.SOURCE\_ADDRESS**

Allow generation of source address elements.

Type: `bool`

Default value: `false`

**subcluster1.ete.TRACE\_OUTPUT**

File to which to write trace byte stream.

Type: `string`

Default value: N/A

**subcluster1.ete.TRCRSRTA\_FORCED\_EXCEP**

TRCRSR.TA value for a forcibly traced exception.

Type: `bool`

Default value: `false`

**subcluster1.ete.TSMARK**

Whether timestamp markers are supported.

Type: `bool`

Default value: `true`

### **subcluster1.force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks**

Force MTE tag accesses to **RAZ/WI** and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

Type: `bool`

Default value: `false`

### **subcluster1.force\_zero\_PSTATE\_PAN**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

### **subcluster1.force\_zero\_mpam\_partid\_and\_pmg**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAMO\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

Type: `uint8_t`

Default value: `0`

### **subcluster1.has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

### **subcluster1.has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

### **subcluster1.has\_enhanced\_pan**

Implements Armv8.7 Enhanced PAN feature (FEAT\_PAN3) values of this parameter are:- 1, feature is implemented if ARMv8.7 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 2

### **subcluster1.has\_ete**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (-plugin or -P).

Type: bool

Default value: false

### **subcluster1.has\_large\_va**

Implement support for the extended 52-bit virtual addresses from ARMv8.2 (FEAT\_LVA).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.2 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

### **subcluster1.has\_statistical\_profiling**

Whether Statistical Based Profiling is implemented (FEAT\_SPE).

Type: bool

Default value: false

### **subcluster1.icache-hit\_latency**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster1.icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster1.icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster1.icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

Type: bool

Default value: false

### **subcluster1.icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster1.icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster1.icache-size**

L1 I-Cache size in bytes.

Type: uint32\_t

Default value: 0x10000

### **subcluster1.instruction\_tlb\_size**

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

Type: uint32\_t

Default value: 0x0

### **subcluster1.invalidate\_code\_cache\_on\_icache\_cmo**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.



Type: `uint8_t`

Default value: 0

### **subcluster1.memory\_tagging\_support\_level**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

Type: `uint8_t`

Default value: 3

### **subcluster1.pmu-num\_counters**

Number of PMU counters implemented.

Type: `uint8_t`

Default value: 6

### **subcluster1.ptw\_latency**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

### **subcluster1.stage12\_tlb\_size**

Number of stage1+2 tlb entries. Valid values are 0 or  $\geq 4$ .

Type: `uint32_t`

Default value: 0x80

### **subcluster1.tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

### **subcluster1.tlbi\_stall\_enabled**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: `bool`

Default value: `false`

**subcluster1.treat\_PAC\_as\_NOP**

Non-architecture parameter. Treat Pointer Authentication as **NOP**. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are **NOP**. Branch, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

Type: `bool`

Default value: `false`

**subcluster1.walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

## 3.54 ARMCortexA520CT\_CortexA725CT\_CortexX925CT

Defined in `LISA/ARMCortexA520CT_CortexA725CT_CortexX925CT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
CortexA520 r0p1	Full support
CortexA725 r0p0	Full support
CortexX925 r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following parameters were added:

- `num_acp`
- `subcluster0.has_delayed_dbgreg`
- `subcluster0.has_delayed_sysreg`
- `subcluster1.has_delayed_dbgreg`
- `subcluster1.has_delayed_sysreg`
- `subcluster2.has_delayed_dbgreg`
- `subcluster2.has_delayed_sysreg`

### About ARMCortexA520CT\_CortexA725CT\_CortexX925CT

The number of cores in each subcluster is configurable using the following parameters:

**subcluster0.NUM\_CORES**

Possible values are 1-12 (ARMCortexA520CT).

**subcluster1.NUM\_CORES**

Possible values are 1-12 (ARMCortexA725CT).

**subcluster2.NUM\_CORES**

Possible values are 1-12 (ARMCortexX925CT).

The total number of cores in the cluster cannot exceed 14.

Port arrays are expanded to the width needed for the maximum number of cores in each subcluster. Use the following port array indexes:

- <port\_name>[0-11] for cores in subcluster0.
- <port\_name>[12-23] for cores in subcluster1.
- <port\_name>[24-35] for cores in subcluster2.



**Note**

All instances in the Master cross trigger matrix port array, `cti[36]` must be connected, regardless of the `NUM_CORES` value used.

Core-specific parameters have the following prefixes:

- `subcluster0.cpu0` to `subcluster0.cpu11` identify cores in subcluster0.
- `subcluster1.cpu0` to `subcluster1.cpu11` identify cores in subcluster1.
- `subcluster2.cpu0` to `subcluster2.cpu11` identify cores in subcluster2.

For information about the cores in this model, see:

- [ARMCortexA520CT](#).
- [ARMCortexA725CT](#).
- [ARMCortexX925CT](#).

## Iris and MTI instances for ARMCortexA520CT\_CortexA725CT\_CortexX925CT

This model has the following Iris instances:

Name	Instance type
ARMCortexA520CT_CortexA725CT_CortexX925CT	<a href="#">Cluster_ARM_Cortex-A520_CortexA725_CortexX925_Heterogeneous</a>
ARMCortexA520CT_CortexA725CT_CortexX925CT.AMU	<a href="#">PVBUSLogger</a>
ARMCortexA520CT_CortexA725CT_CortexX925CT.AMU.mapper	<a href="#">PVBUSMapper</a>
ARMCortexA520CT_CortexA725CT_CortexX925CT.DAP	<a href="#">PVBUSLogger</a>
ARMCortexA520CT_CortexA725CT_CortexX925CT.DAP.mapper	<a href="#">PVBUSMapper</a>
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU	<a href="#">DSU-120</a>

Name	Instance type
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.PPU_cluster	PPUV1
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.PPU_coreU (where $U = 0-2$ )	PPUV1
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.PPU_coreU.busslave (where $U = 0-2$ )	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.shared_cache	PVCache
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.shared_cache.upstream[U] (where $U = 0-8$ )	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.MMAP	PVBusLogger
ARMCortexA520CT_CortexA725CT_CortexX925CT.MMAP.mapper	PVBusMapper
ARMCortexA520CT_CortexA725CT_CortexX925CT.RAS	PVBusLogger
ARMCortexA520CT_CortexA725CT_CortexX925CT.RAS.mapper	PVBusMapper
ARMCortexA520CT_CortexA725CT_CortexX925CT.cpuU.debug_rom (where $U = 0-2$ )	debug_rom
ARMCortexA520CT_CortexA725CT_CortexX925CT.ext_bus	PVBusLogger
ARMCortexA520CT_CortexA725CT_CortexX925CT.ext_bus.mapper	PVBusMapper
ARMCortexA520CT_CortexA725CT_CortexX925CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA520CT_CortexA725CT_CortexX925CT.global_debug_rom	debug_rom
ARMCortexA520CT_CortexA725CT_CortexX925CT.secondary_debug_rom	debug_rom
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster0	Subcluster_ARM_Cortex-A520
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster0.cpu0	ARM_Cortex-A520
ARMCortexA520CT_CortexA725CT_CortexX925CT.subclusterU.cpu0.UTLB (where $U = 0-2$ )	TLB
ARMCortexA520CT_CortexA725CT_CortexX925CT.subclusterU.cpuV.dtlb (where $U = 0-2$ ; $V = 0-2$ )	TLB
ARMCortexA520CT_CortexA725CT_CortexX925CT.subclusterU.cpu0.l1dcache (where $U = 0-2$ )	PVCache
ARMCortexA520CT_CortexA725CT_CortexX925CT.subclusterU.cpu0.l1dcache.downstream[0].pvbusmaster (where $U = 0-2$ )	PVBusMaster
ARMCortexA520CT_CortexA725CT_CortexX925CT.subclusterU.cpu0.l1dcache.upstream[0] (where $U = 0-2$ )	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.subclusterU.cpu0.l1icache (where $U = 0-2$ )	PVCache
ARMCortexA520CT_CortexA725CT_CortexX925CT.subclusterU.cpu0.l1icache.downstream[0].pvbusmaster (where $U = 0-2$ )	PVBusMaster
ARMCortexA520CT_CortexA725CT_CortexX925CT.subclusterU.cpu0.l1icache.upstream[0] (where $U = 0-2$ )	PVBusSlave

Name	Instance type
ARMCortexA520CT_CortexA725CT_CortexX925CT.subclusterU.cpu0.l2cache (where $U = 0-2$ )	PVCache
ARMCortexA520CT_CortexA725CT_CortexX925CT.subclusterU.cpu0.l2cache.downstream[0].pvbusmaster (where $U = 0-2$ )	PVBusMaster
ARMCortexA520CT_CortexA725CT_CortexX925CT.subclusterU.cpu0.l2cache.upstream[A] (where $U = 0-2$ ; $A = 0-1$ )	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.subclusterU.sve (where $U = 0-2$ )	ScalableVectorExtension
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster1	Subcluster_ARM_Cortex-A725
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster1.cpu0	ARM_Cortex-A725
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster2	Subcluster_ARM_Cortex-X925
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster2.cpu0	ARM_Cortex-X925

This model has the following MTI trace components:

Name	Component type
ARMCortexA520CT_CortexA725CT_CortexX925CT.AMU	PVBusLogger
ARMCortexA520CT_CortexA725CT_CortexX925CT.AMU.mapper	PVBusMapper
ARMCortexA520CT_CortexA725CT_CortexX925CT.DAP	PVBusLogger
ARMCortexA520CT_CortexA725CT_CortexX925CT.DAP.mapper	PVBusMapper
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU	DSU-120
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.PPU_cluster	PPUv1
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.PPU_coreU (where $U = 0-2$ )	PPUv1
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.PPU_coreU.busslave (where $U = 0-2$ )	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.shared_cache	PVCache
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.shared_cache.upstream[U] (where $U = 0-8$ )	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.MMAP	PVBusLogger
ARMCortexA520CT_CortexA725CT_CortexX925CT.MMAP.mapper	PVBusMapper
ARMCortexA520CT_CortexA725CT_CortexX925CT.RAS	PVBusLogger
ARMCortexA520CT_CortexA725CT_CortexX925CT.RAS.mapper	PVBusMapper
ARMCortexA520CT_CortexA725CT_CortexX925CT.ext_bus	PVBusLogger
ARMCortexA520CT_CortexA725CT_CortexX925CT.ext_bus.mapper	PVBusMapper
ARMCortexA520CT_CortexA725CT_CortexX925CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster0.cpu0	ARM_Cortex-A520
ARMCortexA520CT_CortexA725CT_CortexX925CT.subclusterU.cpu0.UTLB (where $U = 0-2$ )	TLB

Name	Component type
ARMCortexA520CT_CortexA725CT_CortexX925CT.subclusterU.cpu0.l1dcache (where $U = 0-2$ )	PVCache
ARMCortexA520CT_CortexA725CT_CortexX925CT.subclusterU.cpu0.l1dcache.downstream[0].pvbusmaster (where $U = 0-2$ )	PVBusMaster
ARMCortexA520CT_CortexA725CT_CortexX925CT.subclusterU.cpu0.l1dcache.upstream[0] (where $U = 0-2$ )	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.subclusterU.cpu0.l1icache (where $U = 0-2$ )	PVCache
ARMCortexA520CT_CortexA725CT_CortexX925CT.subclusterU.cpu0.l1icache.downstream[0].pvbusmaster (where $U = 0-2$ )	PVBusMaster
ARMCortexA520CT_CortexA725CT_CortexX925CT.subclusterU.cpu0.l1icache.upstream[0] (where $U = 0-2$ )	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.subclusterU.cpu0.l2cache (where $U = 0-2$ )	PVCache
ARMCortexA520CT_CortexA725CT_CortexX925CT.subclusterU.cpu0.l2cache.downstream[0].pvbusmaster (where $U = 0-2$ )	PVBusMaster
ARMCortexA520CT_CortexA725CT_CortexX925CT.subclusterU.cpu0.l2cache.upstream[A] (where $U = 0-2$ ; $A = 0-1$ )	PVBusSlave
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster1.cpu0	ARM_Cortex-A725
ARMCortexA520CT_CortexA725CT_CortexX925CT.subcluster2.cpu0	ARM_Cortex-X925

### Ports for ARMCortexA520CT\_CortexA725CT\_CortexX925CT

Port	Direction	Protocol	Description
acp_s	slave	PVBus	AXI ACP slave port.
AEND0MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND2MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND3MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART2MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	slave	Signal	CHI defined pins.
broadcastcachemaint	slave	Signal	ACE defined pins.
broadcastouter	slave	Signal	ACE defined pins.
broadcastpersist	slave	Signal	CHI defined pins.
cfgend	slave	Signal	This signal is for EE bit initialisation.

Port	Direction	Protocol	Description
cfgsdisable	slave	Signal	This signal disables write access to some secure Interrupt Controller registers.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcs_m_pchannel	master	PChannel	Cluster PCSM signal
cluster_powerdown_out	master	Signal	DEPRECATED - An output from PPU that informs thermal controller of the cluster power info
cluster_ppu_hw_stat	master	Value	Cluster PPU power state
clusterid	slave	Value	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set ManagerID64 of the core.
clusterpmuirq	master	Signal	DynamiQ pmu irq
CNTHPIRQ	master	Signal	Timer signals to SOC.
CNTHPSIRQ	master	Signal	Timer signals to SOC
CNTHVIRQ	master	Signal	Timer signals to SOC.
CNTHVSIRQ	master	Signal	Timer signals to SOC
CNTPNSIRQ	master	Signal	Timer signals to SOC.
CNTPSIRQ	master	Signal	Timer signals to SOC.
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.
CNTVIRQ	master	Signal	Timer signals to SOC.
commirq	master	Signal	Interrupt signal from debug communications channel.
core_clk_in	slave	ClockSignal	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
core_pcs_m_pchannel	master	PChannel	Core PCSM signals
core_powerdown_out	master	Signal	DEPRECATED - An output from PPU that informs thermal controller of the core power info
core_ppu_hw_stat	master	Value	Notify the power state change inside the PPU.
coreerrirq	master	Signal	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq	master	Signal	Fault indicator for a detected 1 or 2 bit ECC error
coreinstrrun	master	Signal	Core Running State
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
ctidbgirq	master	Signal	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	slave	Signal	External debug interface.

Port	Direction	Protocol	Description
dbgnopwrdown	master	Signal	No power-down request.
dbgpwrupreq	master	Signal	Debug power up request.
defaultMP	slave	Signal	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main master interface ports.
dev_debug_s	slave	PVBus	External debug interface.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.
gicreset	master	Signal	An output from PPU that can be used to reset GICCLK domain components
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	slave	PVBus	External debug interface.
pmbirq	master	Signal	Interrupt signal from the statistical profiling unit.
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	master	Signal	PPU cluster interrupt.
ppu_cluster_isolate	master	Signal	PPU Cluster Domain isolation control.
ppu_cluster_wakerequest	slave	Signal	PPU cluster wake request signal.
ppu_core_irq	master	Signal	PPU core interrupt.
ppu_core_isolate	master	Signal	PPU Core Domain isolation control.
ppu_core_wakerequest	slave	Signal	PPU core wake request signal.
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	master	PVBus	The core will generate bus requests on this port.
pvbus_periph_m	master	PVBus	The core can generate peripheral bus request on this port.
rei	slave	Signal	Per core RAM Error Interrupt.
reset	slave	Signal	Reset signal to cluster.
rvbaraddr	slave	Value_64	Reset vector base address.
sci_m	master	SystemCoherencyInterface	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei	slave	Signal	Per core System Error physical pins.
spiden	slave	Signal	External debug interface.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq	master	Signal	Interrupt signal from the trace buffer unit.
utility_bus_s	slave	PVBus	Utility bus slave
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfiq	slave	Signal	Virtualised FIQ.



Port	Direction	Protocol	Description
virq	slave	Signal	Virtualised IRQ.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Per core virtual System Error physical pins.

## Parameters for ARM CortexA520CT\_CortexA725CT\_CortexX925CT

### **AEND0\_DEFAULT**

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

Type: uint64\_t

Default value: 0x0

### **AEND1\_DEFAULT**

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

Type: uint64\_t

Default value: 0x0

### **AEND2\_DEFAULT**

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

Type: uint64\_t

Default value: 0x0

### **AEND3\_DEFAULT**

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

Type: uint64\_t

Default value: 0x0

### **ASTART0\_DEFAULT**

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

Type: uint64\_t

Default value: 0x0

#### **ASTART1\_DEFAULT**

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

Type: `uint64_t`

Default value: 0x0

#### **ASTART2\_DEFAULT**

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

Type: `uint64_t`

Default value: 0x0

#### **ASTART3\_DEFAULT**

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

Type: `uint64_t`

Default value: 0x0

#### **BROADCASTATOMIC**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`

Default value: `true`

#### **BROADCASTCACHEMAIN**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

Type: `bool`

Default value: `false`

#### **BROADCASTOUTER**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

Type: `bool`

Default value: `false`

### **BROADCASTPERSIST**

Enable broadcasting of cache clean to the point of persistence operations. The `broadcastpersist` signal will override this value if used.

Type: `bool`

Default value: `true`

### **CLUSTER\_ID**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the ManagerID64 of the core.

Type: `uint32_t`

Default value: `0x0`

### **CMO\_broadcast\_when\_cache\_state\_modelling\_disabled**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. This value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain even when cache state modelling is disabled. Possible values are:- 0 = All CMOs are broadcast if architecturally required- 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required.

Type: `uint8_t`

Default value: `1`

### **GICDISABLE**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`

Default value: `true`

### **bus\_type**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

Type: `uint8_t`

Default value: `0`

**core\_power\_on\_by\_default**

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPUs which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

Type: `bool`

Default value: `false`

**dcache-state\_modelled**

Set whether D-cache has stateful implementation.

Type: `bool`

Default value: `false`

**default\_opmode**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

Type: `uint8_t`

Default value: 4

**diagnostics**

Enable DynamIQ diagnostic messages.

Type: `bool`

Default value: `false`

**enable\_simulation\_performance\_optimizations**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of `stage12_tlb_size` parameter to 1024).

Type: `bool`

Default value: `true`

**has\_peripheral\_port**

If true, an additional AXI peripheral port is configured.

Type: `bool`

Default value: `false`

**l1cache-state\_modelled**

Set whether L1-cache has stateful implementation.

Type: `bool`

Default value: `false`

**l3cache-has\_mpam**

L3 Cache has MPAM support.

Type: `bool`

Default value: `true`

**l3cache-hit\_latency**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-maintenance\_latency**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-miss\_latency**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-read\_access\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size `l3cache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-read\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l3cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**l3cache-size**

L3 Cache size in bytes.

Type: `uint32_t`

Default value: 0x80000

**l3cache-snoop\_data\_transfer\_latency**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**l3cache-snoop\_issue\_latency**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**l3cache-ways**

L3 Cache number of ways (sets are implicit from size).

Type: `uint16_t`

Default value: 16

**l3cache-write\_access\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size `l3cache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **l3cache-write\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. l3cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **mpmm\_accumulator\_multiplier**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator.value of n means the accumulator will use (n \* accumulator value) to calculate the mpmm threshold (MPMM).is provided as a fast model workaround to handle cases where execution of quantum in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

Type: uint8\_t

Default value: 1

### **num\_acp**

Number of ACP ports.

Type: uint8\_t

Default value: 0

### **subcluster0.CPUCFR**

Value of CPU Configuration Register.

Type: uint64\_t

Default value: 0x0

### **subcluster0.NUM\_CORES**

Number of cores in the subcluster. Total number of cores in cluster may not exceed 14.

Type: uint8\_t

Default value: 1

### **subcluster0.core\_cache\_protection**

core\_cache\_protection can change ERROFR, ERROPFGF and ERROPFGCTL fields. Possible values are: -1:Not implemented (by default), 0:Disabled, 1:Enabled.

Type: int8\_t

Default value: 1

**subcluster0.cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: uint32\_t

Default value: 0x1

**subcluster0.cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: uint32\_t

Default value: 0x1

**subcluster0.cpu0.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: false

**subcluster0.cpu0.CRYPTODISABLE**

Disable cryptographic features.

Type: bool

Default value: false

**subcluster0.cpu0.RVBARADDR**

Value of RVBAR\_ELx register.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter trace\_special\_hlt\_imm16.

Type: bool

Default value: false



**subcluster0.cpu0.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu0.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu0.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu0.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu0.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu0.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu0.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu0.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu0.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu0.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

### **`subcluster0.cpu0.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

### **`subcluster0.cpu0.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **`subcluster0.cpu0.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **`subcluster0.cpu0.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster0.cpu0.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu0.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu0.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster0.cpu0.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu0.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu0.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu0.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu1.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu1.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster0.cpu1.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu1.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu1.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu1.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu1.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu1.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster0.cpu1.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu1.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu1.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu1.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu1.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu1.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu1.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000



**subcluster0.cpu1.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu1.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu1.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster0.cpu1.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster0.cpu1.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu1.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu1.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu1.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu1.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster0.cpu1.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu1.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu1.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster0.cpu1.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu10.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu10.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu10.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu10.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu10.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu10.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu10.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu10.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu10.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu10.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu10.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu10.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu10.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu10.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu10.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu10.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu10.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu10.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu10.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu10.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu10.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu10.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu10.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster0.cpu10.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster0.cpu10.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster0.cpu10.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster0.cpu10.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu10.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu10.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu10.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu10.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu10.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu10.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu10.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu11.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu11.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu11.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu11.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`



**subcluster0.cpu11.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu11.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu11.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu11.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu11.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu11.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu11.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

#### **subcluster0.cpu11.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu11.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu11.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

#### **subcluster0.cpu11.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu11.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu11.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu11.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu11.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu11.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu11.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu11.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu11.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**`subcluster0.cpu11.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**`subcluster0.cpu11.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**`subcluster0.cpu11.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**`subcluster0.cpu11.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**`subcluster0.cpu11.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: 0x0

**`subcluster0.cpu11.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: 0xf000000

**`subcluster0.cpu11.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster0.cpu11.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu11.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu11.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu11.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu2.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu2.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster0.cpu2.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu2.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu2.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu2.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu2.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu2.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster0.cpu2.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu2.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu2.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu2.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu2.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu2.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu2.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000



**subcluster0.cpu2.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu2.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu2.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster0.cpu2.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster0.cpu2.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu2.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu2.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu2.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu2.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster0.cpu2.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu2.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu2.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster0.cpu2.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu3.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu3.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu3.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu3.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu3.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu3.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu3.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu3.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu3.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu3.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu3.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu3.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu3.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu3.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu3.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu3.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster0.cpu3.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster0.cpu3.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster0.cpu3.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster0.cpu3.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu3.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu3.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu3.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu3.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu3.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster0.cpu3.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu3.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster0.cpu4.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu4.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu4.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu4.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`



**subcluster0.cpu4.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu4.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu4.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu4.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu4.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu4.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu4.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu4.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu4.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu4.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**`subcluster0.cpu4.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**`subcluster0.cpu4.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**`subcluster0.cpu4.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**`subcluster0.cpu4.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**`subcluster0.cpu4.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**`subcluster0.cpu4.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**`subcluster0.cpu4.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

#### **`subcluster0.cpu4.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

#### **`subcluster0.cpu4.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster0.cpu4.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu4.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster0.cpu5.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu5.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu5.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu5.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu5.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu5.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu5.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu5.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster0.cpu5.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu5.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu5.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu5.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu5.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu5.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu5.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000



**subcluster0.cpu5.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu5.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu5.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster0.cpu5.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster0.cpu5.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu5.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu5.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu5.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu5.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster0.cpu5.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu5.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu5.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster0.cpu5.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu6.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu6.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu6.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu6.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu6.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu6.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu6.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu6.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu6.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu6.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu6.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu6.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu6.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu6.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu6.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu6.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu6.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster0.cpu6.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster0.cpu6.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster0.cpu6.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster0.cpu6.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu6.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu6.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu6.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu6.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu6.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster0.cpu6.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu6.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster0.cpu7.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu7.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu7.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu7.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`



**subcluster0.cpu7.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu7.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu7.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu7.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu7.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu7.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu7.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu7.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu7.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu7.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**`subcluster0.cpu7.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**`subcluster0.cpu7.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**`subcluster0.cpu7.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**`subcluster0.cpu7.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**`subcluster0.cpu7.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: 0x0

**`subcluster0.cpu7.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: 0xf000000

**`subcluster0.cpu7.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster0.cpu7.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu7.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu7.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu7.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu8.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu8.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu8.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu8.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu8.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu8.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu8.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu8.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster0.cpu8.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu8.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu8.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu8.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu8.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu8.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu8.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000



**subcluster0.cpu8.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu8.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu8.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster0.cpu8.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster0.cpu8.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu8.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu8.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu8.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu8.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu8.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu8.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu8.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu8.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster0.cpu8.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **subcluster0.cpu9.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **subcluster0.cpu9.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **subcluster0.cpu9.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **subcluster0.cpu9.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **subcluster0.cpu9.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **subcluster0.cpu9.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu9.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-read\_bus\_width\_in\_bytes**

L2 Cache read bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x10

**subcluster0.cpu9.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster0.cpu9.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster0.cpu9.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-write\_bus\_width\_in\_bytes**

L2 Cache write bus width in bytes used to calculate per-access timing annotations.

Type: uint32\_t

Default value: 0x20

**subcluster0.cpu9.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu9.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu9.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu9.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu9.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu9.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster0.cpu9.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster0.cpu9.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster0.cpu9.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster0.cpu9.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu9.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu9.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu9.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu9.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu9.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster0.cpu9.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu9.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster0.dcache-hit_latency`**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.dcache-maintenance_latency`**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.dcache-miss_latency`**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.dcache-prefetch_enabled`**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

Type: `bool`



Default value: `false`

### **`subcluster0.dcache-read_access_latency`**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.dcache-read_latency`**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.dcache-size`**

L1 D-Cache size in bytes.

Type: `uint32_t`

Default value: `0x8000`

### **`subcluster0.dcache-snoop_data_transfer_latency`**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.dcache-write_access_latency`**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.ete.CLAIMTAGS**

Number of claim tags.

Type: uint8\_t

Default value: 4

**subcluster0.ete.MAX\_INST\_PER\_Q**

Maximum limit for the number of instructions implied by a Q element.

Type: uint16\_t

Default value: 0x1

**subcluster0.ete.PIDR\_CM0D**

TRCPIDR CM0D value.

Type: uint8\_t

Default value: 0

**subcluster0.ete.PIDR\_REVAND**

TRCPIDR REVAND value.

Type: uint8\_t

Default value: 0

**subcluster0.ete.PIDR\_REVISION**

TRCPIDR REVISION value.

Type: uint8\_t

Default value: 0

**subcluster0.ete.Q\_CADENCE**

Number of instruction blocks traced between two Q elements.

Type: `uint16_t`

Default value: `0x1`

#### **`subcluster0.ete.RES0_STATEFUL`**

Whether **RES0** bits are stateful or **RAZ/WI**.

Type: `bool`

Default value: `false`

#### **`subcluster0.ete.RETSTACK`**

Return stack depth.

Type: `uint8_t`

Default value: `3`

#### **`subcluster0.ete.REVISION`**

TRCIDR1 revision value.

Type: `uint8_t`

Default value: `0`

#### **`subcluster0.ete.SIM_OVERFLOW_GRANULARITY`**

Number of instruction blocks in each granule, for simulated overflow.

Type: `uint32_t`

Default value: `0x64`

#### **`subcluster0.ete.SIM_OVERFLOW_PERCENTAGE`**

Percentage of instruction blocks lost in each granule, for simulated overflow.

Type: `uint8_t`

Default value: `0`

#### **`subcluster0.ete.SOURCE_ADDRESS`**

Allow generation of source address elements.

Type: `bool`

Default value: `false`

#### **`subcluster0.ete.TRACE_OUTPUT`**

File to which to write trace byte stream.

Type: `string`

Default value: `N/A`

#### **`subcluster0.ete.TRCRSRTA_FORCED_EXCEP`**

TRCRSR.TA value for a forcibly traced exception.

Type: `bool`

Default value: `false`

#### **`subcluster0.force_mte_tag_access_razwi_and_ignore_tag_checks`**

Force MTE tag accesses to **RAZ/WI** and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

Type: `bool`

Default value: `false`

#### **`subcluster0.force_zero_PSTATE_PAN`**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

#### **`subcluster0.force_zero_mpam_partid_and_pmg`**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

Type: `uint8_t`

Default value: `0`

#### **`subcluster0.has_actlr2`**

If true ACLTR2 exists and ACTLR2(NS) is aliased to ACTLR\_EL1[63:32].

Type: `bool`

Default value: `true`

#### **`subcluster0.has_delayed_dbgreg`**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

### **`subcluster0.has_delayed_sysreg`**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

### **`subcluster0.has_dot_product`**

Possible values of this parameter are:- 1, feature not implemented.- 2, feature is implemented.

Type: `uint8_t`

Default value: 2

### **`subcluster0.has_ete`**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (-plugin or -P).

Type: `bool`

Default value: `false`

### **`subcluster0.icache-hit_latency`**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.icache-maintenance_latency`**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.icache-miss_latency`**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **subcluster0.icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

Type: bool

Default value: false

### **subcluster0.icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster0.icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster0.icache-size**

L1 I-Cache size in bytes.

Type: uint32\_t

Default value: 0x8000

### **subcluster0.invalidate\_code\_cache\_on\_icache\_cmo**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: uint8\_t

Default value: 0

**subcluster0.memory\_tagging\_support\_level**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

Type: uint8\_t

Default value: 3

**subcluster0.pmu-num\_counters**

Number of PMU counters implemented.

Type: uint8\_t

Default value: 6

**subcluster0.ptw\_latency**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

**subcluster0.ras\_extra\_configurations**

Miscellaneous configurations for error records. An array of JSON objects. Note for ERXCTLR\_EL1 register it only allows to define the mask value for the IMPDEF fields, ie bits [63:32] and bit 1, but its reset value applies on all fields. Note for ERXMISCN masks - these are 64 bit masks covering the 64 bit registers ERXMISCN\_EL1. E.g. [{"Index": 0, "ERXMISCO\_mask": 0x0, "ERXMISCO\_reset": 0x0, "ERXMISC1\_mask": 0x0, "ERXMISC1\_reset": 0x0, "ERXMISC2\_mask": 0x0, "ERXMISC2\_reset": 0x0, "ERXMISC3\_mask": 0x0, "ERXMISC3\_reset": 0x0, "ERXCTLR\_EL1\_mask": 0x0, "ERXCTLR\_EL1\_reset": 0x0}, {"Index": 1, "ERXMISCO\_mask": 0x0, "ERXMISCO\_reset": 0x0, "ERXSTATUS\_IERR\_mask": 0x300}].

Type: string

Default value: "[ { \"Index\": 1, \"ERXMISCO\_mask\": 0xFFFFc0003fc3, \"ERXMISC1\_mask\": 0x03F870003FF30f07, \"ERXPFGCTL\_reset\": 0x1000 }, { \"Index\": 2, \"ERXMISCO\_mask\": 0xFFFFFe007ffc0, \"ERXMISCO\_reset\": 0x2, \"ERXSTATUS\_IERR\_mask\": 0x300 , \"ERXMISC1\_mask\": 0x0FF8700ffff31f0f, \"ERXPFGCTL\_reset\": 0x1000 } ]"

**subcluster0.ras\_pfg\_clock\_mhz**

RAS Pseudo-Fault generation clock rate in MHz.

Type: uint8\_t

Default value: 12

**subcluster0.revision\_number**

Revision number of TRM implemented by the model. Changing the value can change the model behaviour. Visible as the Variant field in MIDR/MIDR\_EL1. Corresponds to the revision number X in rXpY.

Type: `uint8_t`

Default value: 0

**subcluster0.tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

**subcluster0.tlbi\_stall\_enabled**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: `bool`

Default value: `false`

**subcluster0.treat\_PAC\_as\_NOP**

Non-architecture parameter. Treat Pointer Authentication as **NOP**. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are **NOP**. Branch, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

Type: `bool`

Default value: `false`

**subcluster0.walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

**subcluster1.CPUCFR**

Value of CPU Configuration Register.

Type: `uint64_t`

Default value: 0x0



**subcluster1.NUM\_CORES**

Number of cores in the subcluster. Total number of cores in cluster may not exceed 14.

Type: `uint8_t`

Default value: 1

**subcluster1.cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: 0x1

**subcluster1.cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: 0x1

**subcluster1.cpu0.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu0.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster1.cpu0.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu0.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: 0x0

**subcluster1.cpu0.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter trace\_special\_hlt\_imm16.

Type: bool

Default value: false

**subcluster1.cpu0.force-fpsid**

Override the FPSID value.

Type: bool

Default value: true

**subcluster1.cpu0.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu0.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu0.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu0.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu0.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu0.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x100000

#### **subcluster1.cpu0.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu0.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu0.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu0.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu0.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

### **`subcluster1.cpu0.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

### **`subcluster1.cpu0.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu0.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu0.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

### **`subcluster1.cpu0.semihosting-T32_HLT`**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster1.cpu0.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster1.cpu0.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster1.cpu0.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster1.cpu0.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu0.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu0.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu0.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu0.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu0.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu0.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu0.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster1.cpu1.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu1.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster1.cpu1.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu1.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu1.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu1.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu1.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu1.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu1.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu1.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu1.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu1.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x100000`

#### **`subcluster1.cpu1.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu1.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`



**subcluster1.cpu1.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu1.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu1.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu1.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu1.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu1.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu1.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu1.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster1.cpu1.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster1.cpu1.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu1.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu1.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu1.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu1.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu1.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu1.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu1.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu1.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu1.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster1.cpu10.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu10.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster1.cpu10.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu10.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu10.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu10.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu10.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu10.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu10.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu10.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu10.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu10.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x100000`

**subcluster1.cpu10.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu10.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu10.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu10.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu10.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu10.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu10.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu10.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu10.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu10.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster1.cpu10.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster1.cpu10.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu10.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu10.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: true

**subcluster1.cpu10.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu10.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu10.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster1.cpu10.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu10.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu10.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.



Type: `bool`

Default value: `false`

### **`subcluster1.cpu10.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster1.cpu11.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster1.cpu11.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu11.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu11.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu11.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu11.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu11.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu11.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu11.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu11.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu11.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

#### **subcluster1.cpu11.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x100000

#### **subcluster1.cpu11.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu11.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu11.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu11.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu11.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu11.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu11.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu11.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu11.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu11.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster1.cpu11.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster1.cpu11.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster1.cpu11.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster1.cpu11.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu11.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu11.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu11.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu11.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu11.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu11.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu11.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster1.cpu2.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster1.cpu2.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu2.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu2.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter trace\_special\_hlt\_imm16.

Type: `bool`

Default value: `false`

**subcluster1.cpu2.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu2.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu2.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x100000

**subcluster1.cpu2.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu2.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu2.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu2.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t



Default value: 0x0

**subcluster1.cpu2.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu2.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu2.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu2.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu2.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu2.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster1.cpu2.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

### **`subcluster1.cpu2.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **`subcluster1.cpu2.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **`subcluster1.cpu2.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster1.cpu2.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu2.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster1.cpu2.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster1.cpu2.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

#### **`subcluster1.cpu2.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster1.cpu2.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu2.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster1.cpu3.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu3.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu3.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu3.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu3.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu3.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x100000

**subcluster1.cpu3.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu3.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu3.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

### **`subcluster1.cpu3.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

### **`subcluster1.cpu3.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu3.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu3.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu3.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster1.cpu3.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster1.cpu3.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster1.cpu3.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster1.cpu3.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu3.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu3.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu3.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu3.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu3.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu3.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster1.cpu4.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`



**subcluster1.cpu4.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster1.cpu4.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu4.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu4.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu4.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu4.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu4.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu4.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu4.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu4.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu4.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x100000`

### **`subcluster1.cpu4.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu4.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu4.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu4.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu4.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu4.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu4.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu4.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu4.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu4.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster1.cpu4.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster1.cpu4.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu4.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu4.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu4.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu4.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu4.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster1.cpu4.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu4.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu4.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster1.cpu4.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster1.cpu5.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu5.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu5.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu5.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu5.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu5.force-fpsid`**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu5.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu5.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu5.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu5.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu5.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu5.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x100000

**subcluster1.cpu5.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu5.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu5.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu5.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu5.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu5.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t



Default value: 0

**subcluster1.cpu5.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster1.cpu5.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster1.cpu5.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: 0x123456

**subcluster1.cpu5.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster1.cpu5.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster1.cpu5.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster1.cpu5.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster1.cpu5.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: true

**subcluster1.cpu5.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu5.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu5.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster1.cpu5.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu5.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu5.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu5.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster1.cpu6.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu6.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster1.cpu6.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu6.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu6.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu6.force-fpsid`**

Override the FPSID value.

Type: `bool`

Default value: `true`

### **`subcluster1.cpu6.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu6.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu6.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu6.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu6.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to

the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu6.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x100000`

#### **`subcluster1.cpu6.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu6.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu6.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu6.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu6.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu6.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu6.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu6.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu6.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu6.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster1.cpu6.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster1.cpu6.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu6.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu6.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu6.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu6.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu6.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu6.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: 0xf000000

**subcluster1.cpu6.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu6.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster1.cpu6.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: true

**subcluster1.cpu7.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: false

**subcluster1.cpu7.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: bool

Default value: false

**subcluster1.cpu7.CRYPTODISABLE**

Disable cryptographic features.

Type: bool

Default value: false



**subcluster1.cpu7.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu7.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu7.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu7.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu7.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu7.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu7.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu7.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu7.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x100000

**subcluster1.cpu7.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu7.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu7.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu7.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu7.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

### **`subcluster1.cpu7.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

### **`subcluster1.cpu7.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu7.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu7.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu7.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster1.cpu7.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster1.cpu7.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster1.cpu7.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster1.cpu7.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu7.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu7.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu7.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu7.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu7.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu7.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu7.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster1.cpu8.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu8.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster1.cpu8.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu8.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu8.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu8.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu8.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu8.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu8.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu8.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu8.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu8.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x100000`

### **`subcluster1.cpu8.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu8.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu8.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu8.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu8.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu8.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu8.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000



**subcluster1.cpu8.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu8.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu8.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster1.cpu8.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster1.cpu8.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu8.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu8.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu8.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu8.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu8.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu8.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu8.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu8.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu8.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster1.cpu9.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster1.cpu9.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu9.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu9.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu9.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu9.force-fpsid`**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster1.cpu9.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x100000

#### **subcluster1.cpu9.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu9.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu9.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu9.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu9.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

#### **subcluster1.cpu9.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu9.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster1.cpu9.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster1.cpu9.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: 0x123456

**subcluster1.cpu9.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster1.cpu9.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster1.cpu9.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster1.cpu9.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster1.cpu9.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: true

**subcluster1.cpu9.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu9.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster1.cpu9.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu9.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu9.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu9.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster1.dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.dcache-maintenance\_latency**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.dcache-miss\_latency**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.dcache-prefetch\_enabled**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

Type: `bool`

Default value: `false`



**subcluster1.dcache-read\_access\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.dcache-size**

L1 D-Cache size in bytes.

Type: uint32\_t

Default value: 0x10000

**subcluster1.dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.dcache-write\_access\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.ete.CLAIMTAGS**

Number of claim tags.

Type: uint8\_t

Default value: 4

**subcluster1.ete.ETE\_REVISION**

ETE revision: 0=ETEv1.0, 1=ETEv1.1, 2=ETEv1.2, 3=ETEv1.3.

Type: uint8\_t

Default value: 1

**subcluster1.ete.MAX\_INST\_PER\_Q**

Maximum limit for the number of instructions implied by a Q element.

Type: uint16\_t

Default value: 0x1

**subcluster1.ete.NumberOfRSPairs**

Number of resource selector pairs.

Type: uint8\_t

Default value: 8

**subcluster1.ete.PIDR\_CM0D**

TRCPIDR CM0D value.

Type: uint8\_t

Default value: 0

**subcluster1.ete.PIDR\_REVAND**

TRCPIDR REVAND value.

Type: `uint8_t`

Default value: 0

#### **`subcluster1.ete.PIDR_REVISION`**

TRCPIDR REVISION value.

Type: `uint8_t`

Default value: 0

#### **`subcluster1.ete.Q_CADENCE`**

Number of instruction blocks traced between two Q elements.

Type: `uint16_t`

Default value: 0x1

#### **`subcluster1.ete.RES0_STATEFUL`**

Whether **RES0** bits are stateful or **RAZ/WI**.

Type: `bool`

Default value: `false`

#### **`subcluster1.ete.RETSTACK`**

Return stack depth.

Type: `uint8_t`

Default value: 1

#### **`subcluster1.ete.REVISION`**

TRCIDR1 revision value.

Type: `uint8_t`

Default value: 0

#### **`subcluster1.ete.SIM_OVERFLOW_GRANULARITY`**

Number of instruction blocks in each granule, for simulated overflow.

Type: `uint32_t`

Default value: 0x64

#### **`subcluster1.ete.SIM_OVERFLOW_PERCENTAGE`**

Percentage of instruction blocks lost in each granule, for simulated overflow.

Type: `uint8_t`

Default value: 0

#### **`subcluster1.ete.SOURCE_ADDRESS`**

Allow generation of source address elements.

Type: `bool`

Default value: `false`

#### **`subcluster1.ete.TRACE_OUTPUT`**

File to which to write trace byte stream.

Type: `string`

Default value: N/A

#### **`subcluster1.ete.TRCRSRTA_FORCED_EXCEP`**

TRCRSR.TA value for a forcibly traced exception.

Type: `bool`

Default value: `false`

#### **`subcluster1.ete.TSMARK`**

Whether timestamp markers are supported.

Type: `bool`

Default value: `true`

#### **`subcluster1.force_mte_tag_access_razwi_and_ignore_tag_checks`**

Force MTE tag accesses to **RAZ/WI** and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

Type: `bool`

Default value: `false`

#### **`subcluster1.force_zero_PSTATE_PAN`**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

### **`subcluster1.force_zero_mpam_partid_and_pmg`**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAMO\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

Type: `uint8_t`

Default value: `0`

### **`subcluster1.has_delayed_dbgreg`**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

### **`subcluster1.has_delayed_sysreg`**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

### **`subcluster1.has_enhanced_pan`**

Implements Armv8.7 Enhanced PAN feature (FEAT\_PAN3) values of this parameter are:- 1, feature is implemented if ARMv8.7 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: `2`

### **`subcluster1.has_ete`**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (`-plugin` or `-P`).

Type: `bool`

Default value: `false`

### **`subcluster1.has_large_va`**

Implement support for the extended 52-bit virtual addresses from ARMv8.2 (FEAT\_LVA). values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.2 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

**subcluster1.has\_statistical\_profiling**

Whether Statistical Based Profiling is implemented (FEAT\_SPE).

Type: bool

Default value: false

**subcluster1.icache-hit\_latency**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

Type: bool

Default value: false

**subcluster1.icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.icache-read_latency`**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.icache-size`**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: `0x10000`

### **`subcluster1.instruction_tlb_size`**

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

Type: `uint32_t`

Default value: `0x0`

### **`subcluster1.invalidate_code_cache_on_icache_cmo`**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: `uint8_t`

Default value: `0`

### **`subcluster1.memory_tagging_support_level`**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

Type: `uint8_t`

Default value: `3`

### **`subcluster1.pmu-num_counters`**

Number of PMU counters implemented.

Type: `uint8_t`

Default value: 6

### **`subcluster1.ptw_latency`**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

### **`subcluster1.stage12_tlb_size`**

Number of stage1+2 tlb entries. Valid values are 0 or  $\geq 4$ .

Type: `uint32_t`

Default value: 0x80

### **`subcluster1.tlb_latency`**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

### **`subcluster1.tlbi_stall_enabled`**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: `bool`

Default value: `false`

### **`subcluster1.treat_PAC_as_NOP`**

Non-architecture parameter. Treat Pointer Authentication as **NOP**. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are **NOP**. Branch, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

Type: `bool`

Default value: `false`

### **`subcluster1.walk_cache_latency`**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`



Default value: 0x0

**subcluster2.CPUCFR**

Value of CPU Configuration Register.

Type: uint64\_t

Default value: 0x0

**subcluster2.NUM\_CORES**

Number of cores in the subcluster. Total number of cores in cluster may not exceed 14.

Type: uint8\_t

Default value: 1

**subcluster2.cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: uint32\_t

Default value: 0x1

**subcluster2.cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: uint32\_t

Default value: 0x1

**subcluster2.cpu0.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: false

**subcluster2.cpu0.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: bool

Default value: false

**subcluster2.cpu0.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu0.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu0.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu0.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu0.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu0.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu0.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu0.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu0.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

### **`subcluster2.cpu0.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu0.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu0.l2cache-ways`**

L2 Cache number of ways (sets are implicit from size).

Type: `uint8_t`

Default value: `8`

### **`subcluster2.cpu0.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu0.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu0.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

### **`subcluster2.cpu0.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

### **`subcluster2.cpu0.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster2.cpu0.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster2.cpu0.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster2.cpu0.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster2.cpu0.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster2.cpu0.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster2.cpu0.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster2.cpu0.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster2.cpu0.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu0.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu0.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster2.cpu0.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu0.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster2.cpu0.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster2.cpu0.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster2.cpu1.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster2.cpu1.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster2.cpu1.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster2.cpu1.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu1.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster2.cpu1.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu1.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu1.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu1.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu1.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu1.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

### **`subcluster2.cpu1.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu1.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`



**subcluster2.cpu1.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: `uint8_t`

Default value: 8

**subcluster2.cpu1.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster2.cpu1.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster2.cpu1.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: 0x100

**subcluster2.cpu1.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: 0

**subcluster2.cpu1.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster2.cpu1.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster2.cpu1.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster2.cpu1.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster2.cpu1.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster2.cpu1.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster2.cpu1.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster2.cpu1.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster2.cpu1.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu1.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu1.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster2.cpu1.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu1.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster2.cpu1.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster2.cpu1.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster2.cpu10.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster2.cpu10.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

#### **`subcluster2.cpu10.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster2.cpu10.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster2.cpu10.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

#### **`subcluster2.cpu10.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu10.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu10.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu10.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu10.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu10.l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

**subcluster2.cpu10.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster2.cpu10.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu10.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster2.cpu10.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu10.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu10.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster2.cpu10.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster2.cpu10.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster2.cpu10.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster2.cpu10.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: 0x123456

**subcluster2.cpu10.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster2.cpu10.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster2.cpu10.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster2.cpu10.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster2.cpu10.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: true

**subcluster2.cpu10.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu10.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster2.cpu10.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster2.cpu10.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster2.cpu10.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000



**subcluster2.cpu10.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster2.cpu10.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster2.cpu11.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster2.cpu11.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster2.cpu11.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster2.cpu11.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu11.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu11.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu11.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu11.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu11.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu11.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu11.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster2.cpu11.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu11.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu11.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster2.cpu11.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu11.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu11.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster2.cpu11.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster2.cpu11.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu11.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu11.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster2.cpu11.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster2.cpu11.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster2.cpu11.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster2.cpu11.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster2.cpu11.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster2.cpu11.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu11.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu11.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster2.cpu11.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: 0xf000000

**subcluster2.cpu11.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu11.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster2.cpu11.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: true

**subcluster2.cpu2.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: false

**subcluster2.cpu2.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: bool

Default value: false

**subcluster2.cpu2.CRYPTODISABLE**

Disable cryptographic features.

Type: bool

Default value: false

**subcluster2.cpu2.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu2.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster2.cpu2.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu2.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu2.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu2.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu2.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu2.l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

**subcluster2.cpu2.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu2.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu2.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: `uint8_t`

Default value: `8`

**subcluster2.cpu2.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`



**subcluster2.cpu2.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu2.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster2.cpu2.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster2.cpu2.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu2.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu2.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster2.cpu2.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

### **`subcluster2.cpu2.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

### **`subcluster2.cpu2.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **`subcluster2.cpu2.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **`subcluster2.cpu2.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster2.cpu2.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu2.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster2.cpu2.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster2.cpu2.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster2.cpu2.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster2.cpu2.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu2.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster2.cpu3.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster2.cpu3.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu3.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu3.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu3.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu3.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu3.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu3.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu3.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu3.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu3.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster2.cpu3.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu3.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu3.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster2.cpu3.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu3.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu3.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster2.cpu3.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster2.cpu3.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu3.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu3.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster2.cpu3.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster2.cpu3.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster2.cpu3.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster2.cpu3.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster2.cpu3.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster2.cpu3.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu3.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu3.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster2.cpu3.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu3.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster2.cpu3.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster2.cpu3.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`



**subcluster2.cpu4.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster2.cpu4.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster2.cpu4.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster2.cpu4.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu4.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster2.cpu4.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu4.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu4.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu4.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu4.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu4.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

### **`subcluster2.cpu4.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu4.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu4.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster2.cpu4.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu4.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu4.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster2.cpu4.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster2.cpu4.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster2.cpu4.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster2.cpu4.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster2.cpu4.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster2.cpu4.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster2.cpu4.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster2.cpu4.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster2.cpu4.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster2.cpu4.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu4.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu4.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster2.cpu4.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu4.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster2.cpu4.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu4.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster2.cpu5.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster2.cpu5.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu5.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu5.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu5.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster2.cpu5.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu5.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu5.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu5.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu5.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu5.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

#### **subcluster2.cpu5.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster2.cpu5.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster2.cpu5.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

#### **subcluster2.cpu5.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster2.cpu5.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster2.cpu5.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t



Default value: 0x100

**subcluster2.cpu5.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster2.cpu5.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu5.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu5.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster2.cpu5.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster2.cpu5.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster2.cpu5.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster2.cpu5.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster2.cpu5.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster2.cpu5.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu5.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu5.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster2.cpu5.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu5.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster2.cpu5.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster2.cpu5.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster2.cpu6.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster2.cpu6.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

#### **`subcluster2.cpu6.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster2.cpu6.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu6.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter trace\_special\_hlt\_imm16.

Type: bool

Default value: false

**subcluster2.cpu6.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu6.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu6.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu6.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu6.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster2.cpu6.l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: 0x80000

### **subcluster2.cpu6.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **subcluster2.cpu6.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **subcluster2.cpu6.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: `uint8_t`

Default value: 8

### **subcluster2.cpu6.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **subcluster2.cpu6.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster2.cpu6.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster2.cpu6.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster2.cpu6.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu6.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu6.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster2.cpu6.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster2.cpu6.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

### **subcluster2.cpu6.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **subcluster2.cpu6.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **subcluster2.cpu6.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **subcluster2.cpu6.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **subcluster2.cpu6.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **subcluster2.cpu6.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **subcluster2.cpu6.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster2.cpu6.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster2.cpu6.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu6.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster2.cpu7.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster2.cpu7.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu7.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`



Default value: `false`

### **`subcluster2.cpu7.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu7.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu7.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu7.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu7.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu7.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

#### **subcluster2.cpu7.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster2.cpu7.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

#### **subcluster2.cpu7.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster2.cpu7.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster2.cpu7.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

#### **subcluster2.cpu7.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster2.cpu7.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster2.cpu7.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

### **subcluster2.cpu7.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

### **subcluster2.cpu7.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

### **subcluster2.cpu7.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

### **subcluster2.cpu7.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster2.cpu7.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster2.cpu7.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster2.cpu7.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster2.cpu7.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster2.cpu7.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster2.cpu7.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: 0x0

**subcluster2.cpu7.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: 0xf000000

**subcluster2.cpu7.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster2.cpu7.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster2.cpu7.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster2.cpu7.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster2.cpu7.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster2.cpu8.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster2.cpu8.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster2.cpu8.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster2.cpu8.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu8.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster2.cpu8.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu8.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu8.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu8.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu8.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu8.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

### **`subcluster2.cpu8.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu8.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu8.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster2.cpu8.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu8.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu8.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster2.cpu8.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster2.cpu8.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000



**subcluster2.cpu8.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster2.cpu8.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster2.cpu8.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster2.cpu8.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster2.cpu8.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster2.cpu8.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster2.cpu8.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster2.cpu8.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu8.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster2.cpu8.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster2.cpu8.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster2.cpu8.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu8.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster2.cpu8.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster2.cpu9.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster2.cpu9.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu9.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu9.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.cpu9.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster2.cpu9.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu9.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu9.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu9.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu9.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.cpu9.l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

**subcluster2.cpu9.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster2.cpu9.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu9.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster2.cpu9.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu9.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu9.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster2.cpu9.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster2.cpu9.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster2.cpu9.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster2.cpu9.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: 0x123456

**subcluster2.cpu9.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster2.cpu9.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster2.cpu9.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster2.cpu9.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster2.cpu9.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: true

**subcluster2.cpu9.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster2.cpu9.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster2.cpu9.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster2.cpu9.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster2.cpu9.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster2.cpu9.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster2.cpu9.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster2.dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.dcache-maintenance\_latency**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.dcache-miss\_latency**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.dcache-prefetch\_enabled**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

Type: `bool`

Default value: `false`



**subcluster2.dcache-read\_access\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.dcache-size**

L1 D-Cache size in bytes.

Type: uint32\_t

Default value: 0x10000

**subcluster2.dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.dcache-write\_access\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster2.ecv\_support\_level**

Implement Enhanced Counter Virtualization feature from ARMv8.6. 0, Not supported. 1, fully supported without CNTPOFF. 2, fully supported with CNTPOFF (FEAT\_ECV).

Type: uint8\_t

Default value: 2

**subcluster2.ete.CLAIMTAGS**

Number of claim tags.

Type: uint8\_t

Default value: 4

**subcluster2.ete.MAX\_INST\_PER\_Q**

Maximum limit for the number of instructions implied by a Q element.

Type: uint16\_t

Default value: 0x1

**subcluster2.ete.NumberOfRSPairs**

Number of resource selector pairs.

Type: uint8\_t

Default value: 8

**subcluster2.ete.PIDR\_CM0D**

TRCPIDR CM0D value.

Type: uint8\_t

Default value: 0

**subcluster2.ete.Q\_CADENCE**

Number of instruction blocks traced between two Q elements.

Type: `uint16_t`

Default value: `0x1`

#### **`subcluster2.ete.RES0_STATEFUL`**

Whether **RES0** bits are stateful or **RAZ/WI**.

Type: `bool`

Default value: `false`

#### **`subcluster2.ete.RETSTACK`**

Return stack depth.

Type: `uint8_t`

Default value: `3`

#### **`subcluster2.ete.SIM_OVERFLOW_GRANULARITY`**

Number of instruction blocks in each granule, for simulated overflow.

Type: `uint32_t`

Default value: `0x64`

#### **`subcluster2.ete.SIM_OVERFLOW_PERCENTAGE`**

Percentage of instruction blocks lost in each granule, for simulated overflow.

Type: `uint8_t`

Default value: `0`

#### **`subcluster2.ete.SOURCE_ADDRESS`**

Allow generation of source address elements.

Type: `bool`

Default value: `false`

#### **`subcluster2.ete.TRACE_OUTPUT`**

File to which to write trace byte stream.

Type: `string`

Default value: `N/A`

#### **`subcluster2.ete.TRCRSRTA_FORCED_EXCEP`**

TRCRSR.TA value for a forcibly traced exception.

Type: `bool`

Default value: `false`

### **`subcluster2.ext_abort_so_write_ras_type`**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `uint8_t`

Default value: 2

### **`subcluster2.force_mte_tag_access_razwi_and_ignore_tag_checks`**

Force MTE tag accesses to **RAZ/WI** and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

Type: `bool`

Default value: `false`

### **`subcluster2.force_zero_PSTATE_PAN`**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

### **`subcluster2.force_zero_mpam_partid_and_pmg`**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

Type: `uint8_t`

Default value: 0

### **`subcluster2.has_delayed_dbgreg`**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

### **`subcluster2.has_delayed_sysreg`**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

### **subcluster2.has\_enhanced\_pan**

Implements Armv8.7 Enhanced PAN feature (FEAT\_PAN3) values of this parameter are:- 1, feature is implemented if ARMv8.7 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 2

### **subcluster2.has\_ete**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (-plugin or -P).

Type: `bool`

Default value: `false`

### **subcluster2.has\_mt\_pmu\_disable\_feature**

Implements Multi-threading PMU disable extension from ARMv8.6 (FEAT\_MTPMU). 0: FEAT\_MTPMU is disabled, 1: FEAT\_MTPMU is enabled if ARMv8.6 is implemented, 2: FEAT\_MTPMU is cherry-picked, 0xF: The feature is disabled and is represented by value 0xF in ID\_AA64DFR0\_EL1.MTPMU.

Type: `uint8_t`

Default value: 0

### **subcluster2.has\_statistical\_profiling**

Whether Statistical Based Profiling is implemented (FEAT\_SPE).

Type: `bool`

Default value: `true`

### **subcluster2.has\_v8\_7\_spe\_inverted\_filtering**

Where FEAT\_SPEv1p2 is implemented, whether inverted filtering by events is implemented (represented by PMISDR.FnE).

Type: `bool`

Default value: `false`

### **subcluster2.has\_v8\_7\_spe\_previous\_branch\_target**

Where FEAT\_SPEv1p2 is implemented, whether the optional branch target feature is implemented (FEAT\_SPE\_PBT).

Type: `bool`

Default value: `false`

### **`subcluster2.icache-hit_latency`**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.icache-maintenance_latency`**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.icache-miss_latency`**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster2.icache-prefetch_enabled`**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

### **`subcluster2.icache-read_access_latency`**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.icache-size**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: `0x10000`

**subcluster2.instruction\_tlb\_size**

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

Type: `uint32_t`

Default value: `0x0`

**subcluster2.invalidate\_code\_cache\_on\_icache\_cmo**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: `uint8_t`

Default value: `0`

**subcluster2.memory\_tagging\_support\_level**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

Type: `uint8_t`

Default value: `3`

**subcluster2.mpam\_has\_altsp**

DEPRECATED: MPAMIDR\_EL1.HAS\_ALTSP is required when FEAT\_RME is implemented.

Type: `bool`

Default value: `false`

**subcluster2.mpamidr\_has\_force\_ns**

Whether MPAMIDR\_EL1.HAS\_FORCE\_NS bit is set or clear.values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.6 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

**subcluster2.mpamidr\_has\_sdeflt**

Whether MPAMIDR\_EL1.HAS\_SDEFLT bit is set or clear.values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.6 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

**subcluster2.mpamidr\_has\_tidr**

Whether MPAMIDR\_EL1.HAS\_TIDR bit is set or clear.values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.6 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

**subcluster2.pmu-num\_counters**

Number of PMU counters implemented.

Type: uint8\_t

Default value: 31

**subcluster2.ptw\_latency**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

**subcluster2.stage12\_tlb\_size**

Number of stage1+2 tlb entries. Valid values are 0 or >= 4.

Type: uint32\_t

Default value: 0x80



**subcluster2.tcr\_txsz\_undersize\_should\_fault**

If large VA is not supported, Whether undersized TxSZ value should generate translation fault.

Type: `bool`

Default value: `false`

**subcluster2.tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

**subcluster2.tlbi\_stall\_enabled**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: `bool`

Default value: `false`

**subcluster2.treat\_PAC\_as\_NOP**

Non-architecture parameter. Treat Pointer Authentication as **NOP**. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are **NOP**. Branch, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

Type: `bool`

Default value: `false`

**subcluster2.walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

## 3.55 ARMCortexA53CT

Defined in `LISA/ARMCortexA53CT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p1	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

## Changes in 11.31.15

The following parameters were added:

- `has_delayed_dbgreg`
- `has_delayed_sysreg`

## About ARM Cortex A53CT

This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (crypto.dll or crypto.so) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

The system designer decides whether a debug APB ties the external debug view with other system views. In the model, use a PVBUSDecoder to direct traffic to the correct port, `dev_debug_s` or `memorymapped_debug_s`.

## Differences between the CT model and RTL implementations

This component has the following differences from the corresponding revision of the RTL implementation:

- The value of the AArch64 `PMCEID0_ELO` register, and the AArch32 alias of this register, differs in the model from the TRM value. The model value reflects the model counters.
- The mechanisms for setting the affinity fields of the `MPIDR`. The RTL has two ports:
  - `CLUSTERIDAFF1[7:0]`.
  - `CLUSTERIDAFF2[7:0]`. `AFF1` sets the value of `MPIDR` bits[15:8] and `AFF2` sets the value of `MPIDR` bits[23:16]. In contrast, the model has a single `CLUSTER_ID` port. This difference allows the setting of bits[23:8] of the `MPIDR` using bits[15:0] of the `CLUSTER_ID` value.
- The memory mapped debug registers have a view for cores and a view for external debug agents. In the model, these views require two PVBUS ports. In hardware, the system designer decides how the implementation differentiates the views.
- In the model, a single peer event port combines the functionality of the `eventi` and `evento` signals in the RTL.
- The Generic Timers are Programmer's View (PV) level abstractions: a model-specific protocol connects the `cntvalueb` port to the MemoryMappedCounterModule.
- The GIC CPU Interface is a PV level abstraction: a model-specific protocol connects the GIC CPU Interface to the GIC Distributor.
- The CoreSight Cross Trigger Interface (CTI) is a PV level abstraction: the interface is a model specific one.
- The model has no mechanism to read the internal memory that the Cache and TLB structures use, through the implementation defined region of the system coprocessor interface. This

memory includes the RAM Index Register, `IL1DATA` Registers, `DL1DATA` Registers, and associated functionality.

- The model does not implement:
  - ETM registers.
  - The `PMUEVENT` bus.
  - The `WARMRESETREQ` signal. However, the warm reset code sequence (see the section [Code sequence to request a Warm reset as a result of `RMR\_ELx.RR` in the \[Arm Architecture Reference Manual for A-profile architecture\]\(#\)](#) ) makes the model simulate a warm reset of the core.
  - The `PMUSNAPSHOTREQ` and `PMUSNAPSHOTACK` signals.
  - The `EXTERRIRQ` and `INTERRIRQ` signals.
  - Processor dynamic-retention signals.
  - The `SYSBARDISABLE` signal.
  - This model has a variable number of cores per cluster, specified using the `NUM_CORES` parameter.
  - The `DBGPWRDUP`, `DBGPWRUPREQ`, `DBGNOPWRDWN`, and `DBGIRSTREQ` debug power management signals.
  - The RTL synthesis option to remove FP and ASE.
  - The RTL synthesis option for a Cortex-A15 style debug memory map.
  - Although Neon support is optional for the Cortex-A53 processor, this model does not implement the `ase-present` parameter. This means it is not possible to configure the model to not support Neon.
  - ECC and parity schemes are hardware-specific so are not supported.

## Iris and MTI instances for ARMCortexA53CT

This model has the following Iris instances:

Name	Instance type
ARMCortexA53CT	<a href="#">Cluster_ARM_Cortex-A53</a>
ARMCortexA53CT.AMU	<a href="#">PVBUSLogger</a>
ARMCortexA53CT.AMU.mapper	<a href="#">PVBUSMapper</a>
ARMCortexA53CT.DAP	<a href="#">PVBUSLogger</a>
ARMCortexA53CT.DAP.mapper	<a href="#">PVBUSMapper</a>
ARMCortexA53CT.DSU	<a href="#">DSU</a>
ARMCortexA53CT.DSU.mpam_busslave	<a href="#">PVBUSSlave</a>
ARMCortexA53CT.MMAP	<a href="#">PVBUSLogger</a>
ARMCortexA53CT.MMAP.mapper	<a href="#">PVBUSMapper</a>
ARMCortexA53CT.RAS	<a href="#">PVBUSLogger</a>
ARMCortexA53CT.RAS.mapper	<a href="#">PVBUSMapper</a>
ARMCortexA53CT.acp_mapper	<a href="#">PVBUSMapper</a>
ARMCortexA53CT.cpu0	<a href="#">ARM_Cortex-A53</a>

Name	Instance type
ARMCortexA53CT.cpu0.UTLB	TLB
ARMCortexA53CT.cpu0.dtlb	TLB
ARMCortexA53CT.cpu0.l1dcache	PVCache
ARMCortexA53CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA53CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA53CT.cpu0.l1icache	PVCache
ARMCortexA53CT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA53CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA53CT.ext_bus	PVBusLogger
ARMCortexA53CT.ext_bus.mapper	PVBusMapper
ARMCortexA53CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA53CT.global_debug_rom	debug_rom
ARMCortexA53CT.l2_cache	PVCache
ARMCortexA53CT.l2_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA53CT.l2_cache.upstream[Z] (where Z = 0-16)	PVBusSlave
ARMCortexA53CT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

Name	Component type
ARMCortexA53CT.AMU	PVBusLogger
ARMCortexA53CT.AMU.mapper	PVBusMapper
ARMCortexA53CT.DAP	PVBusLogger
ARMCortexA53CT.DAP.mapper	PVBusMapper
ARMCortexA53CT.DSU	DSU
ARMCortexA53CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA53CT.MMAP	PVBusLogger
ARMCortexA53CT.MMAP.mapper	PVBusMapper
ARMCortexA53CT.RAS	PVBusLogger
ARMCortexA53CT.RAS.mapper	PVBusMapper
ARMCortexA53CT.acp_mapper	PVBusMapper
ARMCortexA53CT.cpu0	ARM_Cortex-A53
ARMCortexA53CT.cpu0.UTLB	TLB
ARMCortexA53CT.cpu0.l1dcache	PVCache
ARMCortexA53CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA53CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA53CT.cpu0.l1icache	PVCache
ARMCortexA53CT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA53CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA53CT.ext_bus	PVBusLogger
ARMCortexA53CT.ext_bus.mapper	PVBusMapper

Name	Component type
ARMCortexA53CT.gic_cpuif_decoder_cluster	<a href="#">GICv3CPUInterfaceDecoder</a>
ARMCortexA53CT.l2_cache	<a href="#">PVCache</a>
ARMCortexA53CT.l2_cache.downstream[0].pvbusmaster	<a href="#">PVBusMaster</a>
ARMCortexA53CT.l2_cache.upstream[Z] (where Z = 0-16)	<a href="#">PVBusSlave</a>
ARMCortexA53CT.l2_flusher	<a href="#">AsyncCacheFlushUnit</a>

## Ports for ARMCortexA53CT

Port	Direction	Protocol	Description
aa64naa32	slave	<a href="#">Signal</a>	Register width after reset.
acp_s	slave	<a href="#">PVBus</a>	AXI ACP slave port.
broadcastcachemaint	slave	<a href="#">Signal</a>	Enable broadcasting of cache maintenance operations to downstream caches.
broadcastinner	slave	<a href="#">Signal</a>	Enable broadcasting of Inner Shareable transactions.
broadcastouter	slave	<a href="#">Signal</a>	Enable broadcasting of Outer Shareable transactions.
cfgend	slave	<a href="#">Signal</a>	This signal if for EE bit initialisation.
cfgsdisable	slave	<a href="#">Signal</a>	This signal disables write access to some secure Interrupt Controller registers.
cfgte	slave	<a href="#">Signal</a>	This signal provides default exception handling state.
clk_in	slave	<a href="#">ClockSignal</a>	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
clrexmonack	master	<a href="#">Signal</a>	Acknowledge handshake signal for the clrexmonreq signal
clrexmonreq	slave	<a href="#">Signal</a>	Signals the clearing of an external global exclusive monitor
clusterid	slave	<a href="#">Value</a>	The port sets the value of the affinity levels 1 and 2; bits [23:16] and [15:8] of the MPIDR.
CNTHPIRQ	master	<a href="#">Signal</a>	Timer signals to SoC.
CNTPNSIRQ	master	<a href="#">Signal</a>	Timer signals to SoC.
CNTPSIRQ	master	<a href="#">Signal</a>	Timer signals to SoC.
cntvalueb	slave	<a href="#">CounterInterface</a>	Interface to SoC level counter module.
CNTVIRQ	master	<a href="#">Signal</a>	Timer signals to SoC.
commirq	master	<a href="#">Signal</a>	Interrupt signal from debug communications channel.
commr_x	master	<a href="#">Signal</a>	Receive portion of Data Transfer Register full.
commtx	master	<a href="#">Signal</a>	Transmit portion of Data Transfer Register empty.
cp15sdisable	slave	<a href="#">Signal</a>	This signal disables write access to some system control processor registers.
cpuporeset	slave	<a href="#">Signal</a>	Power on reset. Initializes all the processor logic, including debug logic.
cryptodisable	slave	<a href="#">Signal</a>	Disable cryptography extensions after reset.
cti	master	<a href="#">v8EmbeddedCrossTrigger_controlprotocol</a>	Cross trigger matrix port.
ctidbgirq	master	<a href="#">Signal</a>	Cross Trigger Interface (CTI) interrupt trigger output.
dbgack	master	<a href="#">Signal</a>	External debug interface.

Port	Direction	Protocol	Description
dbgen	slave	Signal	External debug interface.
dbgllrstdisable	slave	Signal	Control ram clear on reset
dbgnopwrdown	master	Signal	This signals relate to core power down.
dbgpwrupreq	master	Signal	This signals relate to core power down.
dev_debug_s	slave	PVBus	External debug interface.
edbgrq	slave	Signal	External debug interface.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
l2flushdone	master	Signal	Flush of L2 memory system complete.
l2flushreq	slave	Signal	Request flush of L2 memory system.
l2reset	slave	Signal	Reset the shared L2 memory system controller.
l2rstdisable	slave	Signal	-
memorymapped_debug_s	slave	PVBus	External debug interface.
niden	slave	Signal	External debug interface.
periphbase	slave	Value_64	This port sets the base address of private peripheral region.
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	master	PVBus	The core will generate bus requests on this port.
rei	slave	Signal	Individual processor RAM Error Interrupt signal input.
reset	slave	Signal	Raising this signal will put the core into reset mode.
romaddr	slave	Value_64	Debug ROM base address.
romaddrv	slave	Signal	Debug ROM base address valid.
rvbaraddr	slave	Value_64	Reset vector base address.
sei	slave	Signal	Per core System Error physical pins.
smpen	master	Signal	This signals AMP or SMP mode for each core.
spiden	slave	Signal	External debug interface.
spniden	slave	Signal	External debug interface.
standbywfe	master	Signal	This signal indicates if a core is in WFE state.
standbywfi	master	Signal	This signal indicates if a core is in WFI state.
standbywfil2	master	Signal	Indicate that all the individual processors and the L2 systems are in a WFI state.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfiq	slave	Signal	Virtual FIQ.

Port	Direction	Protocol	Description
vinithi	slave	Signal	This signal controls of the location of the exception vectors at reset.
virq	slave	Signal	Virtual IRQ.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Processor Virtual System Error Interrupt request.

## Parameters for ARMCortexA53CT

### **cpuX.AA64nAA32**

Type: `bool`

Default value: `true`

### **cpuX.CFGEND**

Type: `bool`

Default value: `false`

### **cpuX.CFGTE**

Type: `bool`

Default value: `false`

### **cpuX.CP15SDISABLE**

Type: `bool`

Default value: `false`

### **cpuX.CRYPTODISABLE**

Type: `bool`

Default value: `false`

### **cpuX.RVBARADDR**

Type: `uint64_t`

Default value: `0`

### **cpuX.VINITHI**

Type: `bool`

Default value: `false`

**cpuX.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter trace\_special\_hlt\_imm16.

Type: `bool`

Default value: `false`

**cpuX.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint64_t`

Default value: `256`

**cpuX.min\_sync\_level**

force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint32_t`

Default value: `0`

**cpuX.semihosting-A32\_HLT**

Type: `uint32_t`

Default value: `0xF000`

**cpuX.semihosting-A64\_HLT**

Type: `uint32_t`

Default value: `0xF000`

**cpuX.semihosting-ARM\_SVC**

Type: `uint32_t`

Default value: `0x123456`

**cpuX.semihosting-T32\_HLT**

Type: `uint32_t`

Default value: `0x3c`

**cpuX.semihosting-Thumb\_SVC**

Type: `uint32_t`

Default value: `0xAB`



**cpuX.semihosting-cmd\_line**

Type: string

Default value: ""

**cpuX.semihosting-cwd**

Type: string

Default value: ""

**cpuX.semihosting-enable**

Type: bool

Default value: true

**cpuX.semihosting-heap\_base**

Type: uint32\_t

Default value: 0x0

**cpuX.semihosting-heap\_limit**

Type: uint32\_t

Default value: 0x0F000000

**cpuX.semihosting-stack\_base**

Type: uint32\_t

Default value: 0x10000000

**cpuX.semihosting-stack\_limit**

Type: uint32\_t

Default value: 0x0F000000

**cpuX.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**cpuX.vfp-enable\_at\_reset**

Type: bool

Default value: false

**cpuX.vfp-present**

Set whether the model has VFP support.

Type: `bool`

Default value: `true`

**BROADCASTCACHEMAINT**

Enable broadcasting of cache maintenance operations to downstream caches. The `broadcastcachemaint` signal will override this value if used.

Type: `bool`

Default value: `true`

**BROADCASTINNER**

Enable broadcasting of Inner Shareable transactions. The `broadcastinner` signal will override this value if used.

Type: `bool`

Default value: `true`

**BROADCASTOUTER**

Enable broadcasting of Outer Shareable transactions. The `broadcastouter` signal will override this value if used.

Type: `bool`

Default value: `true`

**CLUSTER\_ID**

Type: `uint32_t`

Default value: `0`

**DBGROMADDR**

Type: `uint64_t`

Default value: `0x0`

**DBGROMADDRV**

Type: `bool`

Default value: `true`

**GICDISABLE**

Type: `bool`

Default value: `true`

**NUM\_CORES**

Number of cores in cluster.

Type: `uint8_t`

Default value: `1`

**PERIPHBASE**

Type: `uint64_t`

Default value: `0x13080000`

**bus\_type**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

Type: `uint32_t`

Default value: `0`

**cpi\_div**

divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `1`

**cpi\_mul**

multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `1`

**dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit.

Type: `uint64_t`

Default value: `0`

**dcache-maintenance\_latency**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

**dcache-miss\_latency**

L1 D-Cache timing annotation latency for miss.

Type: `uint64_t`

Default value: 0

**dcache-prefetch\_enabled**

Enable simulation of data cache prefetching. This is only used when `dcache-dcache_state_modelled=true`.

Type: `bool`

Default value: `false`

**dcache-read\_access\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (based on the size of `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

**dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

**dcache-size**

L1 D-Cache size in bytes.

Type: `uint32_t`

Default value: `0x8000`

**dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0`

**dcache-state\_modelled**

Type: `bool`

Default value: `false`

**dcache-write\_access\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (based on the size of `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0`

**dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0`

**has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**icache-hit\_latency**

L1 I-Cache timing annotation latency for hit.

Type: `uint64_t`

Default value: 0

### **icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

### **icache-miss\_latency**

L1 I-Cache timing annotation latency for miss.

Type: `uint64_t`

Default value: 0

### **icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

### **icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (based on the size of `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

### **icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

### **icache-size**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: `0x8000`

### **`icache-state_modelled`**

Type: `bool`

Default value: `false`

### **`l2cache-hit_latency`**

L2 Cache timing annotation latency for hit.

Type: `uint64_t`

Default value: `0`

### **`l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0`

### **`l2cache-miss_latency`**

L2 Cache timing annotation latency for miss.

Type: `uint64_t`

Default value: `0`

### **`l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access (based on the size of `l2cache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0`

### **`l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0`

**l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

**l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0`

**l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0`

**l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access (based on the size of `l2cache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0`

**l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0`

**patch\_level**

Cosmetic change to Revision field in MIDR/MIDR\_EL1. Corresponds to the patch number Y in rXpY.

Type: `uint32_t`

Default value: `0x1`



**ptw\_latency**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint32_t`

Default value: 0

**revision\_number**

Cosmetic change to Variant field in MIDR/MIDR\_EL1. Corresponds to the revision number X in rXpY.

Type: `uint32_t`

Default value: 0x0

**tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint32_t`

Default value: 0

**walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint32_t`

Default value: 0

## 3.56 ARMCortexA55CT

Defined in `LISA/ARMCortexA55CT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r1p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following parameters were added:

- `has_delayed_dbgreg`
- `has_delayed_sysreg`

## About ARMCortexA55CT

The model supports the following features:

- DynamIQ (DSU) system registers.
- Per-core L2 cache.
- A P-Channel for the cluster and for each core.
- Optional peripheral port.
- L3 cache partition.
- Per-core clock.
- BROADCASTPERSIST pin.

The model does not support the following features:

- BROADCASTCACHEMAINTPOU pin.
- COREINSTRRET, COREINSTRRUN, or nPMBIRQ signals.
- DynamIQ features that are irrelevant to the programmers' view simulation, for example:
  - Automatic CPU retention mode.
  - Level-3 Cache RAM retention.
- 256-bit wide output transactions.
- ECC and parity schemes are hardware-specific so are not supported.
- Self-test features (MBIST).
- Latency configuration.
- Snoop filtering.
- Debug power management signals DBGPWRDUP and DBGIRSTREQ are not implemented, but DBGPWRUPREQ and DBGNOFWRDWN are implemented.
- Cache stashing capability.
- Dual ACE masters.

This model has a variable number of cores per cluster, specified using the `NUM_CORES` parameter.

The per-core parameters are preceded by `cpuN.`, where N identifies the core (0-7).

This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (`crypto.dll` or `crypto.so`) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

## Iris and MTI instances for ARMCortexA55CT

This model has the following Iris instances:

Name	Instance type
ARMCortexA55CT	Cluster_ARM_Cortex-A55

Name	Instance type
ARMCortexA55CT.AMU	PVBusLogger
ARMCortexA55CT.AMU.mapper	PVBusMapper
ARMCortexA55CT.DAP	PVBusLogger
ARMCortexA55CT.DAP.mapper	PVBusMapper
ARMCortexA55CT.DSU	DSU
ARMCortexA55CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA55CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA55CT.DSU.shared_cache	PVCache
ARMCortexA55CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA55CT.DSU.shared_cache.upstream[Y] (where Y = 0-3)	PVBusSlave
ARMCortexA55CT.MMAP	PVBusLogger
ARMCortexA55CT.MMAP.mapper	PVBusMapper
ARMCortexA55CT.RAS	PVBusLogger
ARMCortexA55CT.RAS.mapper	PVBusMapper
ARMCortexA55CT.cpu0	ARM_Cortex-A55
ARMCortexA55CT.cpu0.UTLB	TLB
ARMCortexA55CT.cpu0.dtlb	TLB
ARMCortexA55CT.cpu0.l1dcache	PVCache
ARMCortexA55CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA55CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA55CT.cpu0.l1licache	PVCache
ARMCortexA55CT.cpu0.l1licache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA55CT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA55CT.cpu0.l2cache	PVCache
ARMCortexA55CT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA55CT.cpu0.l2cache.upstream[U] (where U = 0-1)	PVBusSlave
ARMCortexA55CT.ext_bus	PVBusLogger
ARMCortexA55CT.ext_bus.mapper	PVBusMapper
ARMCortexA55CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA55CT.global_debug_rom	debug_rom

This model has the following MTI trace components:

Name	Component type
ARMCortexA55CT.AMU	PVBusLogger
ARMCortexA55CT.AMU.mapper	PVBusMapper
ARMCortexA55CT.DAP	PVBusLogger
ARMCortexA55CT.DAP.mapper	PVBusMapper
ARMCortexA55CT.DSU	DSU
ARMCortexA55CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA55CT.DSU.mpam_busslave	PVBusSlave

Name	Component type
ARMCortexA55CT.DSU.shared_cache	PVCache
ARMCortexA55CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA55CT.DSU.shared_cache.upstream[Y] (where Y = 0–3)	PVBusSlave
ARMCortexA55CT.MMAP	PVBusLogger
ARMCortexA55CT.MMAP.mapper	PVBusMapper
ARMCortexA55CT.RAS	PVBusLogger
ARMCortexA55CT.RAS.mapper	PVBusMapper
ARMCortexA55CT.cpu0	ARM_Cortex-A55
ARMCortexA55CT.cpu0.UTLB	TLB
ARMCortexA55CT.cpu0.l1dcache	PVCache
ARMCortexA55CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA55CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA55CT.cpu0.l1icache	PVCache
ARMCortexA55CT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA55CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA55CT.cpu0.l2cache	PVCache
ARMCortexA55CT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA55CT.cpu0.l2cache.upstream[U] (where U = 0–1)	PVBusSlave
ARMCortexA55CT.ext_bus	PVBusLogger
ARMCortexA55CT.ext_bus.mapper	PVBusMapper
ARMCortexA55CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

## Ports for ARMCortexA55CT

Port	Direction	Protocol	Description
aa64naa32	slave	Signal	Register width after reset.
acp_s	slave	PVBus	AXI ACP slave port.
AENDMP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
ASTARTMP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	slave	Signal	CHI defined pins.
broadcastcachemaint	slave	Signal	ACE defined pins.
broadcastouter	slave	Signal	ACE defined pins.
broadcastpersist	slave	Signal	CHI defined pins.
cfgend	slave	Signal	This signal if for EE bit initialisation.
cfgsdisable	slave	Signal	This signal disables write access to some secure Interrupt Controller registers.
cfgte	slave	Signal	This signal provides default exception handling state.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.

Port	Direction	Protocol	Description
clusterid	slave	Value	The port sets the value of the affinity levels 2 and 3; bits [39:32] and [23:16] of the MPIDR.
clusterpmuirq	master	Signal	DynalQ pmu irq
CNTHPIRQ	master	Signal	Timer signals to SOC.
CNTHVIRQ	master	Signal	Timer signals to SOC.
CNTPNSIRQ	master	Signal	Timer signals to SOC.
CNTPSIRQ	master	Signal	Timer signals to SOC.
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.
CNTVIRQ	master	Signal	Timer signals to SOC.
commirq	master	Signal	Interrupt signal from debug communications channel.
core_clk_in	slave	ClockSignal	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
cp15sdisable	slave	Signal	This signal disables write access to some system control processor registers.
cpuporeset	slave	Signal	CPU power on reset. Initializes all the processor logic, including debug logic.
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
ctidbgirq	master	Signal	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	slave	Signal	External debug interface.
dbgnopwrdown	master	Signal	No power-down request.
dbgpwrupreq	master	Signal	Debug power up request.
dev_debug_s	slave	PVBus	External debug interface.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	slave	PVBus	External debug interface.
niden	slave	Signal	External debug interface.
pchannel_cluster	slave	PChannel	PChannel for cluster.
pchannel_core	slave	PChannel	PChannels for cores
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	master	PVBus	The core will generate bus requests on this port.
pvbus_periph_m	master	PVBus	The core can generate peripheral bus request on this port.
rei	slave	Signal	Per core RAM Error Interrupt.
reset	slave	Signal	Raising this signal will put the core into reset mode.
rvbaraddr	slave	Value_64	Reset vector base address.

Port	Direction	Protocol	Description
sci_m	master	SystemCoherencyInterface	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei	slave	Signal	Per core System Error physical pins.
spiden	slave	Signal	External debug interface.
spniden	slave	Signal	External debug interface.
sporeset	slave	Signal	A single cluster-wide power on reset signal for all resettable registers in DynamIQ.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfiq	slave	Signal	Virtualised FIQ.
vinithi	slave	Signal	This signal controls of the location of the exception vectors at reset.
virq	slave	Signal	Virtualised IRQ.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Per core virtual System Error physical pins.

## Parameters for ARM Cortex A55CT

### **cpuX.AA64nAA32**

Type: `bool`

Default value: `true`

### **cpuX.CFGEND**

Type: `bool`

Default value: `false`

### **cpuX.CFGTE**

Type: `bool`

Default value: `false`

### **cpuX.CP15SDISABLE**

Type: `bool`

Default value: `false`

### **cpuX.CRYPTODISABLE**

Type: `bool`

Default value: `false`

**cpuX.RVBARADDR**

Type: `uint64_t`

Default value: 0

**cpuX.VINITHI**

Type: `bool`

Default value: `false`

**cpuX.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit.

Type: `uint64_t`

Default value: 0

**cpuX.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**cpuX.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss.

Type: `uint64_t`

Default value: 0

**cpuX.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access (based on the size of `l2cache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**cpuX.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x40000

**cpuX.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access (based on the size of l2cache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t



Default value: 0x0

**cpuX.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint64\_t

Default value: 256

**cpuX.min\_sync\_level**

force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint32\_t

Default value: 0

**cpuX.semihosting-A32\_HLT**

Type: uint32\_t

Default value: 0xF000

**cpuX.semihosting-A64\_HLT**

Type: uint32\_t

Default value: 0xF000

**cpuX.semihosting-ARM\_SVC**

Type: uint32\_t

Default value: 0x123456

**cpuX.semihosting-T32\_HLT**

Type: uint32\_t

Default value: 0x3c

**cpuX.semihosting-Thumb\_SVC**

Type: uint32\_t

Default value: 0xAB

**cpuX.semihosting-cmd\_line**

Type: string

Default value: ""

**cpuX.semihosting-cwd**

Type: string

Default value: ""

**cpuX.semihosting-enable**

Type: bool

Default value: true

**cpuX.semihosting-heap\_base**

Type: uint32\_t

Default value: 0x0

**cpuX.semihosting-heap\_limit**

Type: uint32\_t

Default value: 0x0F000000

**cpuX.semihosting-stack\_base**

Type: uint32\_t

Default value: 0x10000000

**cpuX.semihosting-stack\_limit**

Type: uint32\_t

Default value: 0x0F000000

**cpuX.vfp-enable\_at\_reset**

Type: bool

Default value: false

**cpuX.vfp-present**

Set whether the model has VFP support.

Type: bool

Default value: true

**BROADCASTATOMIC**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: bool

Default value: `true`

### **BROADCASTCACHEMAINT**

Enable broadcasting of cache maintenance operations to downstream caches. The `broadcastcachemaint` signal will override this value if used.

Type: `bool`

Default value: `true`

### **BROADCASTOUTER**

Enable broadcasting of Outer Shareable transactions. The `broadcastouter` signal will override this value if used.

Type: `bool`

Default value: `true`

### **BROADCASTPERSIST**

Enable broadcasting of cache clean to the point of persistence operations. The `broadcastpersist` signal will override this value if used.

Type: `bool`

Default value: `true`

### **CLUSTER\_ID**

Type: `uint32_t`

Default value: 0

### **CMO\_broadcast\_when\_cache\_state\_modelling\_disabled**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. This value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain events even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required.

Type: `uint8_t`

Default value: 1

### **GICDISABLE**

Type: `bool`

Default value: `true`

**NUM\_CORES**

Number of cores in cluster.

Type: uint8\_t

Default value: 1

**bus\_type**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

Type: uint32\_t

Default value: 0

**cpi\_div**

divider for calculating CPI (Cycles Per Instruction).

Type: uint32\_t

Default value: 1

**cpi\_mul**

multiplier for calculating CPI (Cycles Per Instruction).

Type: uint32\_t

Default value: 1

**dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit.

Type: uint64\_t

Default value: 0

**dcache-maintenance\_latency**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0

**dcache-miss\_latency**

L1 D-Cache timing annotation latency for miss.

Type: uint64\_t

Default value: 0

**dcache-prefetch\_enabled**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

Type: bool

Default value: false

**dcache-read\_access\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (based on the size of dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0

**dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0

**dcache-size**

L1 D-Cache size in bytes.

Type: uint32\_t

Default value: 0x8000

**dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0

**dcache-state\_modelled**

Type: bool

Default value: false

**dcache-write\_access\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (based on the size of `dcache-write_bus_width_in_bytes`. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

**dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

**enable\_simulation\_performance\_optimizations**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of `stage12_tlb_size` parameter to 1024).

Type: `bool`

Default value: `true`

**force\_zero\_PSTATE\_PAN**

Non-architecture parameter to force `PSTATE.PAN` to be 0. 0: No effect, 1: `PSTATE.PAN` is always treated as 0. The parameter optimizes the performance of updating `PSTATE.PAN`.

Type: `bool`

Default value: `false`

**has\_acp**

If true, Accelerator Coherency Port is configured.

Type: `bool`

Default value: `false`

**has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

**has\_dot\_product**

Possible values of this parameter are:- 1, feature not implemented.- 2, feature is implemented.

Type: `uint8_t`

Default value: 2

**has\_peripheral\_port**

If true, additional AXI peripheral port is configured.

Type: `bool`

Default value: `false`

**icache-hit\_latency**

L1 I-Cache timing annotation latency for hit.

Type: `uint64_t`

Default value: 0

**icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

**icache-miss\_latency**

L1 I-Cache timing annotation latency for miss.

Type: `uint64_t`

Default value: 0

**icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

Type: `bool`

Default value: `false`

**icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (based on the size of icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is only used when icache-state\_modelled=true.

Type: `uint64_t`

Default value: 0

**icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. This is only used when icache-state\_modelled=true.

Type: `uint64_t`

Default value: 0

**icache-size**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: `0x8000`

**icache-state\_modelled**

Type: `bool`

Default value: `false`

**l3cache-hit\_latency**

L3 Cache timing annotation latency for hit.

Type: `uint64_t`

Default value: 0

**l3cache-maintenance\_latency**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks.



Type: `uint64_t`

Default value: 0

### **l3cache-miss\_latency**

L3 Cache timing annotation latency for miss.

Type: `uint64_t`

Default value: 0

### **l3cache-read\_access\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-read_latency` is set.

Type: `uint64_t`

Default value: 0

### **l3cache-read\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed.

Type: `uint64_t`

Default value: 0

### **l3cache-size**

L3 Cache size in bytes.

Type: `uint32_t`

Default value: 0x80000

### **l3cache-snoop\_data\_transfer\_latency**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed.

Type: `uint64_t`

Default value: 0

### **l3cache-snoop\_issue\_latency**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks.

Type: `uint64_t`

Default value: 0

**l3cache-write\_access\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if 32cache-write\_latency is set.

Type: `uint64_t`

Default value: 0

**l3cache-write\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed.

Type: `uint64_t`

Default value: 0

**pchannel\_treat\_simreset\_as\_poreset**

Register core as ON state to cluster with simulation reset.

Type: `bool`

Default value: `false`

**periph\_address\_end**

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

Type: `uint64_t`

Default value: 0x0

**periph\_address\_start**

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

Type: `uint64_t`

Default value: 0x0

**ptw\_latency**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint32_t`

Default value: 0

**tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint32_t`

Default value: 0

**walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

3.57 ARMCortexA55CT\_CortexA75CT

Defined in LISA/ARMCortexA55CT\_CortexA75CT.lisa.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
CortexA55 r1p0	Full support
CortexA75 r3p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

Changes in 11.31.15

The following parameters were added:

- subcluster0.has\_delayed\_dbgreg
- subcluster0.has\_delayed\_sysreg
- subcluster1.has\_delayed\_dbgreg
- subcluster1.has\_delayed\_sysreg

About ARMCortexA55CT\_CortexA75CT

The number of cores in each subcluster is configurable using the following parameters:

**subcluster0.NUM\_CORES**  
Possible values are 1-7 (ARMCortexA55CT).

**subcluster1.NUM\_CORES**  
Possible values are 1-4 (ARMCortexA75CT).

The total number of cores in the cluster cannot exceed 8.

Port arrays are expanded to the width needed for the maximum number of cores in each subcluster. Use the following port array indexes:

- port\_name[0-6] for cores in subcluster0.
- port\_name[7-10] for cores in subcluster1.



All instances in the Master cross trigger matrix port array, `cti[11]` must be connected, regardless of the `num_cores` value used.

Core-specific parameters have the following prefixes:

- `subcluster0.cpu0` to `subcluster0.cpu6` identify cores in `subcluster0`.
- `subcluster1.cpu0` to `subcluster1.cpu3` identify cores in `subcluster1`.

For information about the cores in this model, see:

- [ARMCortexA55CT](#)
- [ARMCortexA75CT](#)

See also [Arm DynamIQ Shared Unit Technical Reference Manual](#).

This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (`crypto.dll` or `crypto.so`) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

## Iris and MTI instances for ARMCortexA55CT\_CortexA75CT

This model has the following Iris instances:

Name	Instance type
ARMCortexA55CT_CortexA75CT	Cluster_ARM_Cortex-A55_Cortex-A75
ARMCortexA55CT_CortexA75CT.AMU	PVBusLogger
ARMCortexA55CT_CortexA75CT.AMU.mapper	PVBusMapper
ARMCortexA55CT_CortexA75CT.DAP	PVBusLogger
ARMCortexA55CT_CortexA75CT.DAP.mapper	PVBusMapper
ARMCortexA55CT_CortexA75CT.DSU	DSU
ARMCortexA55CT_CortexA75CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA55CT_CortexA75CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA55CT_CortexA75CT.DSU.shared_cache	PVCache
ARMCortexA55CT_CortexA75CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA55CT_CortexA75CT.DSU.shared_cache.upstream[Z] (where Z = 0-5)	PVBusSlave
ARMCortexA55CT_CortexA75CT.MMAP	PVBusLogger
ARMCortexA55CT_CortexA75CT.MMAP.mapper	PVBusMapper
ARMCortexA55CT_CortexA75CT.RAS	PVBusLogger
ARMCortexA55CT_CortexA75CT.RAS.mapper	PVBusMapper
ARMCortexA55CT_CortexA75CT.ext_bus	PVBusLogger
ARMCortexA55CT_CortexA75CT.ext_bus.mapper	PVBusMapper
ARMCortexA55CT_CortexA75CT.gic_cpuif_decoder_cluster	GLCv3CPUInterfaceDecoder
ARMCortexA55CT_CortexA75CT.global_debug_rom	debug_rom

Name	Instance type
ARMCortexA55CT_CortexA75CT.subcluster0	Subcluster_ARM_Cortex-A55
ARMCortexA55CT_CortexA75CT.subcluster0.cpu0	ARM_Cortex-A55
ARMCortexA55CT_CortexA75CT.subclusterZ.cpu0.UTLB (where Z = 0-1)	TLB
ARMCortexA55CT_CortexA75CT.subclusterZ.cpuU.dtlb (where Z = 0-1; U = 0-1)	TLB
ARMCortexA55CT_CortexA75CT.subclusterZ.cpu0.l1dcache (where Z = 0-1)	PVCache
ARMCortexA55CT_CortexA75CT.subclusterZ.cpu0.l1dcache.downstream[0].pvbusmaster (where Z = 0-1)	PVBusMaster
ARMCortexA55CT_CortexA75CT.subclusterZ.cpu0.l1dcache.upstream[0] (where Z = 0-1)	PVBusSlave
ARMCortexA55CT_CortexA75CT.subclusterZ.cpu0.l1icache (where Z = 0-1)	PVCache
ARMCortexA55CT_CortexA75CT.subclusterZ.cpu0.l1icache.downstream[0].pvbusmaster (where Z = 0-1)	PVBusMaster
ARMCortexA55CT_CortexA75CT.subclusterZ.cpu0.l1icache.upstream[0] (where Z = 0-1)	PVBusSlave
ARMCortexA55CT_CortexA75CT.subclusterZ.cpu0.l2cache (where Z = 0-1)	PVCache
ARMCortexA55CT_CortexA75CT.subclusterZ.cpu0.l2cache.downstream[0].pvbusmaster (where Z = 0-1)	PVBusMaster
ARMCortexA55CT_CortexA75CT.subclusterZ.cpu0.l2cache.upstream[W] (where Z = 0-1; W = 0-1)	PVBusSlave
ARMCortexA55CT_CortexA75CT.subcluster1	Subcluster_ARM_Cortex-A75
ARMCortexA55CT_CortexA75CT.subcluster1.cpu0	ARM_Cortex-A75

This model has the following MTI trace components:

Name	Component type
ARMCortexA55CT_CortexA75CT.AMU	PVBusLogger
ARMCortexA55CT_CortexA75CT.AMU.mapper	PVBusMapper
ARMCortexA55CT_CortexA75CT.DAP	PVBusLogger
ARMCortexA55CT_CortexA75CT.DAP.mapper	PVBusMapper
ARMCortexA55CT_CortexA75CT.DSU	DSU
ARMCortexA55CT_CortexA75CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA55CT_CortexA75CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA55CT_CortexA75CT.DSU.shared_cache	PVCache
ARMCortexA55CT_CortexA75CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA55CT_CortexA75CT.DSU.shared_cache.upstream[Z] (where Z = 0-5)	PVBusSlave
ARMCortexA55CT_CortexA75CT.MMAP	PVBusLogger
ARMCortexA55CT_CortexA75CT.MMAP.mapper	PVBusMapper
ARMCortexA55CT_CortexA75CT.RAS	PVBusLogger
ARMCortexA55CT_CortexA75CT.RAS.mapper	PVBusMapper
ARMCortexA55CT_CortexA75CT.ext_bus	PVBusLogger
ARMCortexA55CT_CortexA75CT.ext_bus.mapper	PVBusMapper
ARMCortexA55CT_CortexA75CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA55CT_CortexA75CT.subcluster0.cpu0	ARM_Cortex-A55

Name	Component type
ARMCortexA55CT_CortexA75CT.subclusterZ.cpu0.UTLB (where Z = 0-1)	TLB
ARMCortexA55CT_CortexA75CT.subclusterZ.cpu0.l1dcache (where Z = 0-1)	PVCache
ARMCortexA55CT_CortexA75CT.subclusterZ.cpu0.l1dcache.downstream[0].pvbusmaster (where Z = 0-1)	PVBusMaster
ARMCortexA55CT_CortexA75CT.subclusterZ.cpu0.l1dcache.upstream[0] (where Z = 0-1)	PVBusSlave
ARMCortexA55CT_CortexA75CT.subclusterZ.cpu0.l1licache (where Z = 0-1)	PVCache
ARMCortexA55CT_CortexA75CT.subclusterZ.cpu0.l1licache.downstream[0].pvbusmaster (where Z = 0-1)	PVBusMaster
ARMCortexA55CT_CortexA75CT.subclusterZ.cpu0.l1licache.upstream[0] (where Z = 0-1)	PVBusSlave
ARMCortexA55CT_CortexA75CT.subclusterZ.cpu0.l2cache (where Z = 0-1)	PVCache
ARMCortexA55CT_CortexA75CT.subclusterZ.cpu0.l2cache.downstream[0].pvbusmaster (where Z = 0-1)	PVBusMaster
ARMCortexA55CT_CortexA75CT.subclusterZ.cpu0.l2cache.upstream[W] (where Z = 0-1; W = 0-1)	PVBusSlave
ARMCortexA55CT_CortexA75CT.subcluster1.cpu0	ARM_Cortex-A75

### Ports for ARMCortexA55CT\_CortexA75CT

Port	Direction	Protocol	Description
aa64naa32	slave	Signal	Register width after reset.
acp_s	slave	PVBus	AXI ACP slave port
AENDMP	slave	Value_64	DynamiQ port to obtain end address of valid peripheral address range (exclusive).
ASTARTMP	slave	Value_64	DynamiQ port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	slave	Signal	CHI defined pins.
broadcastcachemaint	slave	Signal	ACE defined pins.
broadcastouter	slave	Signal	ACE defined pins.
broadcastpersist	slave	Signal	CHI defined pins.
cfgend	slave	Signal	This signal if for EE bit initialisation.
cfgsdisable	slave	Signal	This signal disables write access to some secure Interrupt Controller registers.
cfgte	slave	Signal	This signal provides default exception handling state.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
clusterid	slave	Value	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set ManagerID64 of the core.

Port	Direction	Protocol	Description
clusterpmuirq	master	Signal	DynamiQ pmu irq
CNTHPIRQ	master	Signal	Timer signals to SOC.
CNTHVIRQ	master	Signal	Timer signals to SOC.
CNTPNSIRQ	master	Signal	Timer signals to SOC.
CNTPSIRQ	master	Signal	Timer signals to SOC.
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.
CNTVIRQ	master	Signal	Timer signals to SOC.
commirq	master	Signal	Interrupt signal from debug communications channel.
core_clk_in	slave	ClockSignal	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
cp15sdisable	slave	Signal	This signal disables write access to some system control processor registers.
cpuporeset	slave	Signal	CPU power on reset. Initializes all the processor logic, including debug logic.
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
ctidbgirq	master	Signal	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	slave	Signal	External debug interface.
dbgnopwrdown	master	Signal	No power-down request.
dbgppwrupreq	master	Signal	Debug power up request.
dev_debug_s	slave	PVBus	External debug interface.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	slave	PVBus	External debug interface.
niden	slave	Signal	External debug interface.
pchannel_cluster	slave	PChannel	PChannel for cluster.
pchannel_core	slave	PChannel	PChannels for cores
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	master	PVBus	The core will generate bus requests on this port.
pvbus_periph_m	master	PVBus	The core can generate peripheral bus request on this port.
rei	slave	Signal	Per core RAM Error Interrupt.
reset	slave	Signal	Raising this signal will put the core into reset mode.
rvbaraddr	slave	Value_64	Reset vector base address.
sci_m	master	SystemCoherencyInterface	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain

Port	Direction	Protocol	Description
sei	slave	Signal	Per core System Error physical pins.
spiden	slave	Signal	External debug interface.
spniden	slave	Signal	External debug interface.
sporeset	slave	Signal	A single cluster-wide power on reset signal for all resettable registers in DynamIQ.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfiq	slave	Signal	Virtualised FIQ.
vinithi	slave	Signal	This signal controls of the location of the exception vectors at reset.
virq	slave	Signal	Virtualised IRQ.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L3 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Per core virtual System Error physical pins.

## Parameters for ARM CortexA55CT\_CortexA75CT

### BROADCASTATOMIC

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`

Default value: `true`

### BROADCASTCACHEMAINT

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

Type: `bool`

Default value: `true`

### BROADCASTOUTER

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

Type: `bool`

Default value: `true`



**BROADCASTPERSIST**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: `bool`

Default value: `true`

**CLUSTER\_ID**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the ManagerID64 of the core.

Type: `uint32_t`

Default value: `0x0`

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. This value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required.

Type: `uint8_t`

Default value: `1`

**GICDISABLE**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`

Default value: `true`

**bus\_type**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

Type: `uint8_t`

Default value: `0`

**dcache-state\_modelled**

Set whether D-cache has stateful implementation.

Type: `bool`

Default value: `false`

**default\_opmode**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

Type: `uint8_t`

Default value: 4

**diagnostics**

Enable DynamIQ diagnostic messages.

Type: `bool`

Default value: `false`

**enable\_simulation\_performance\_optimizations**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of `stage12_tlb_size` parameter to 1024).

Type: `bool`

Default value: `true`

**has\_acp**

If true, Accelerator Coherency Port is configured.

Type: `bool`

Default value: `false`

**has\_peripheral\_port**

If true, additional AXI peripheral port is configured.

Type: `bool`

Default value: `false`

**icache-state\_modelled**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

**l3cache-hit\_latency**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-maintenance\_latency**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-miss\_latency**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-read\_access\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size `l3cache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-read\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l3cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-size`**

L3 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

### **`l3cache-snoop_data_transfer_latency`**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-snoop_issue_latency`**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-write_access_latency`**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size `l3cache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-write_latency`**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l3cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`pchannel_treat_simreset_as_poreset`**

Register core as ON state to cluster with simulation reset.

Type: `bool`

Default value: `false`

### **`periph_address_end`**

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

Type: `uint64_t`

Default value: `0x0`

### **`periph_address_start`**

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.NUM_CORES`**

Number of cores per cluster.

Type: `uint8_t`

Default value: `1`

### **`subcluster0.cpi_div`**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

### **`subcluster0.cpi_mul`**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

### **`subcluster0.cpu0.AA64nAA32`**

Register width configuration at reset. 0, AArch32. 1, AArch64.

Type: `bool`

Default value: `true`

**subcluster0.cpu0.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster0.cpu0.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster0.cpu0.CP15SDISABLE**

Initialize to disable access to some CP15 registers.

Type: `bool`

Default value: `false`

**subcluster0.cpu0.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster0.cpu0.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu0.VINITHI**

Reset value of SCTLR.V.

Type: `bool`

Default value: `false`

**subcluster0.cpu0.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu0.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu0.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu0.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu0.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu0.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu0.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x40000`

#### **`subcluster0.cpu0.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu0.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu0.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu0.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu0.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

#### **`subcluster0.cpu0.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).



Type: `uint8_t`

Default value: 0

**`subcluster0.cpu0.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**`subcluster0.cpu0.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**`subcluster0.cpu0.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: 0x123456

**`subcluster0.cpu0.semihosting-T32_HLT`**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**`subcluster0.cpu0.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**`subcluster0.cpu0.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**`subcluster0.cpu0.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

### **`subcluster0.cpu0.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster0.cpu0.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu0.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu0.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster0.cpu0.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu0.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu0.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster0.cpu0.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster0.cpu1.AA64nAA32**

Register width configuration at reset. 0, AArch32. 1, AArch64.

Type: `bool`

Default value: `true`

**subcluster0.cpu1.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster0.cpu1.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster0.cpu1.CP15SDISABLE**

Initialize to disable access to some CP15 registers.

Type: `bool`

Default value: `false`

**subcluster0.cpu1.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu1.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu1.VINITHI`**

Reset value of SCTLR.V.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu1.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu1.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu1.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu1.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu1.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x40000

**subcluster0.cpu1.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu1.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu1.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

### **`subcluster0.cpu1.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

### **`subcluster0.cpu1.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu1.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu1.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu1.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster0.cpu1.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster0.cpu1.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster0.cpu1.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster0.cpu1.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu1.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: 0x0

**subcluster0.cpu1.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: 0xf000000

**subcluster0.cpu1.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu1.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu1.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu1.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster0.cpu1.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster0.cpu2.AA64nAA32**

Register width configuration at reset. 0, AArch32. 1, AArch64.

Type: `bool`

Default value: `true`



**subcluster0.cpu2.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster0.cpu2.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster0.cpu2.CP15SDISABLE**

Initialize to disable access to some CP15 registers.

Type: `bool`

Default value: `false`

**subcluster0.cpu2.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster0.cpu2.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu2.VINITHI**

Reset value of SCTLR.V.

Type: `bool`

Default value: `false`

**subcluster0.cpu2.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu2.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu2.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu2.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu2.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu2.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu2.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x40000`

### **`subcluster0.cpu2.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu2.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu2.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu2.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu2.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

### **`subcluster0.cpu2.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: 0

### **`subcluster0.cpu2.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

### **`subcluster0.cpu2.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

### **`subcluster0.cpu2.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: 0x123456

### **`subcluster0.cpu2.semihosting-T32_HLT`**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

### **`subcluster0.cpu2.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

### **`subcluster0.cpu2.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **`subcluster0.cpu2.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

### **`subcluster0.cpu2.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster0.cpu2.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu2.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu2.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster0.cpu2.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu2.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu2.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster0.cpu2.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster0.cpu3.AA64nAA32**

Register width configuration at reset. 0, AArch32. 1, AArch64.

Type: `bool`

Default value: `true`

**subcluster0.cpu3.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster0.cpu3.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster0.cpu3.CP15SDISABLE**

Initialize to disable access to some CP15 registers.

Type: `bool`

Default value: `false`

**subcluster0.cpu3.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu3.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu3.VINITHI`**

Reset value of SCTLR.V.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu3.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu3.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu3.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu3.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu3.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x40000

**subcluster0.cpu3.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.



Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu3.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu3.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

### **`subcluster0.cpu3.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

### **`subcluster0.cpu3.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu3.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu3.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu3.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster0.cpu3.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster0.cpu3.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster0.cpu3.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster0.cpu3.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu3.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: 0x0

**subcluster0.cpu3.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: 0xf000000

**subcluster0.cpu3.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu3.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu3.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu3.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster0.cpu3.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster0.cpu4.AA64nAA32**

Register width configuration at reset. 0, AArch32. 1, AArch64.

Type: `bool`

Default value: `true`

**subcluster0.cpu4.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster0.cpu4.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster0.cpu4.CP15SDISABLE**

Initialize to disable access to some CP15 registers.

Type: `bool`

Default value: `false`

**subcluster0.cpu4.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster0.cpu4.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu4.VINITHI**

Reset value of SCTLR.V.

Type: `bool`

Default value: `false`

**subcluster0.cpu4.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu4.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu4.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu4.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu4.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu4.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu4.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x40000`

#### **`subcluster0.cpu4.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu4.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu4.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu4.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu4.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

#### **`subcluster0.cpu4.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: 0

**`subcluster0.cpu4.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**`subcluster0.cpu4.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**`subcluster0.cpu4.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: 0x123456

**`subcluster0.cpu4.semihosting-T32_HLT`**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**`subcluster0.cpu4.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**`subcluster0.cpu4.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**`subcluster0.cpu4.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

### **`subcluster0.cpu4.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster0.cpu4.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu4.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu4.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster0.cpu4.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu4.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`



**subcluster0.cpu4.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster0.cpu4.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster0.cpu5.AA64nAA32**

Register width configuration at reset. 0, AArch32. 1, AArch64.

Type: `bool`

Default value: `true`

**subcluster0.cpu5.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster0.cpu5.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster0.cpu5.CP15SDISABLE**

Initialize to disable access to some CP15 registers.

Type: `bool`

Default value: `false`

**subcluster0.cpu5.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu5.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu5.VINITHI`**

Reset value of SCTLR.V.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu5.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu5.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu5.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu5.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu5.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x40000

**subcluster0.cpu5.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu5.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu5.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

### **`subcluster0.cpu5.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

### **`subcluster0.cpu5.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu5.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu5.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu5.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster0.cpu5.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster0.cpu5.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster0.cpu5.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster0.cpu5.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu5.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: 0x0

**subcluster0.cpu5.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: 0xf000000

**subcluster0.cpu5.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu5.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu5.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu5.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster0.cpu5.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster0.cpu6.AA64nAA32**

Register width configuration at reset. 0, AArch32. 1, AArch64.

Type: `bool`

Default value: `true`

**subcluster0.cpu6.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster0.cpu6.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster0.cpu6.CP15SDISABLE**

Initialize to disable access to some CP15 registers.

Type: `bool`

Default value: `false`

**subcluster0.cpu6.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster0.cpu6.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu6.VINITHI**

Reset value of SCTLR.V.

Type: `bool`

Default value: `false`

**subcluster0.cpu6.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu6.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu6.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu6.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu6.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu6.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu6.l2cache-size`**

L2 Cache size in bytes.



Type: `uint32_t`

Default value: `0x40000`

### **`subcluster0.cpu6.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu6.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu6.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu6.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu6.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

### **`subcluster0.cpu6.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: 0

#### **`subcluster0.cpu6.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

#### **`subcluster0.cpu6.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

#### **`subcluster0.cpu6.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: 0x123456

#### **`subcluster0.cpu6.semihosting-T32_HLT`**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

#### **`subcluster0.cpu6.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

#### **`subcluster0.cpu6.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

#### **`subcluster0.cpu6.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

### **`subcluster0.cpu6.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster0.cpu6.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu6.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu6.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster0.cpu6.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu6.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu6.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster0.cpu6.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster0.dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.dcache-maintenance\_latency**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.dcache-miss\_latency**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.dcache-prefetch\_enabled**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

Type: `bool`

Default value: `false`

**subcluster0.dcache-read\_access\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.dcache-size**

L1 D-Cache size in bytes.

Type: uint32\_t

Default value: 0x8000

**subcluster0.dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.dcache-write\_access\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.force\_zero\_PSTATE\_PAN**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: bool

Default value: false

**subcluster0.has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: bool

Default value: false

**subcluster0.has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: bool

Default value: true

**subcluster0.has\_dot\_product**

Possible values of this parameter are:- 1, feature not implemented.- 2, feature is implemented.

Type: uint8\_t

Default value: 2

**subcluster0.icache-hit\_latency**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

**subcluster0.icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.icache-size**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: `0x8000`

#### **`subcluster0.invalidate_code_cache_on_icache_cmo`**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: `uint8_t`

Default value: `0`

#### **`subcluster0.ptw_latency`**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.tlb_latency`**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.tlbi_stall_enabled`**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: `bool`

Default value: `false`

#### **`subcluster0.walk_cache_latency`**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.CCSIDR-L1D_override`**

If nonzero, override the value presented in CCSIDR for L1D (this is cosmetic and does not affect cache behaviour).

Type: `uint64_t`

Default value: `0x0`



**subcluster1.CCSIDR-L1I\_override**

If nonzero, override the value presented in CCSIDR for L1I (this is cosmetic and does not affect cache behaviour).

Type: `uint64_t`

Default value: `0x0`

**subcluster1.CCSIDR-L2\_override**

If nonzero, override the value presented in CCSIDR for L2 (this is cosmetic and does not affect cache behaviour).

Type: `uint64_t`

Default value: `0x0`

**subcluster1.NUM\_CORES**

Number of cores per cluster.

Type: `uint8_t`

Default value: `1`

**subcluster1.cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**subcluster1.cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**subcluster1.cpu0.AA64nAA32**

Register width configuration at reset. 0, AArch32. 1, AArch64.

Type: `bool`

Default value: `true`

**subcluster1.cpu0.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster1.cpu0.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu0.CP15SDISABLE`**

Initialize to disable access to some CP15 registers.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu0.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu0.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu0.VINITHI`**

Reset value of SCTLR.V.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu0.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu0.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu0.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu0.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu0.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu0.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu0.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster1.cpu0.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu0.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu0.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu0.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu0.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu0.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu0.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster1.cpu0.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster1.cpu0.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: 0x123456

**subcluster1.cpu0.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster1.cpu0.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster1.cpu0.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster1.cpu0.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster1.cpu0.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: true

**subcluster1.cpu0.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu0.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu0.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster1.cpu0.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu0.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu0.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu1.AA64nAA32**

Register width configuration at reset. 0, AArch32. 1, AArch64.

Type: `bool`

Default value: `true`

**subcluster1.cpu1.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu1.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster1.cpu1.CP15SDISABLE**

Initialize to disable access to some CP15 registers.

Type: `bool`

Default value: `false`

**subcluster1.cpu1.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu1.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.VINITHI`**

Reset value of SCTLR.V.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu1.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu1.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.



Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

### **`subcluster1.cpu1.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu1.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu1.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu1.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu1.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu1.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu1.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu1.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

### **`subcluster1.cpu1.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

### **`subcluster1.cpu1.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **`subcluster1.cpu1.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **`subcluster1.cpu1.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster1.cpu1.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster1.cpu1.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster1.cpu1.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster1.cpu1.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu1.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu2.AA64nAA32`**

Register width configuration at reset. 0, AArch32. 1, AArch64.

Type: `bool`

Default value: `true`

### **`subcluster1.cpu2.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster1.cpu2.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster1.cpu2.CP15SDISABLE**

Initialize to disable access to some CP15 registers.

Type: `bool`

Default value: `false`

**subcluster1.cpu2.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu2.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.VINITHI**

Reset value of SCTLR.V.

Type: `bool`

Default value: `false`

**subcluster1.cpu2.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu2.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu2.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu2.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu2.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu2.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

### **`subcluster1.cpu2.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu2.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu2.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu2.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu2.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu2.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu2.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu2.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu2.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster1.cpu2.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster1.cpu2.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu2.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu2.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`



**subcluster1.cpu2.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu2.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu2.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu2.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu2.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu3.AA64nAA32**

Register width configuration at reset. 0, AArch32. 1, AArch64.

Type: `bool`

Default value: `true`

### **`subcluster1.cpu3.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster1.cpu3.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu3.CP15SDISABLE`**

Initialize to disable access to some CP15 registers.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu3.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu3.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu3.VINITHI`**

Reset value of SCTLR.V.

Type: `bool`

Default value: `false`

**subcluster1.cpu3.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter trace\_special\_hlt\_imm16.

Type: `bool`

Default value: `false`

**subcluster1.cpu3.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: 0x0

### **subcluster1.cpu3.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

### **subcluster1.cpu3.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster1.cpu3.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster1.cpu3.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster1.cpu3.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster1.cpu3.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu3.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu3.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu3.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu3.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu3.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster1.cpu3.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster1.cpu3.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster1.cpu3.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster1.cpu3.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu3.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu3.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu3.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu3.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster1.cpu3.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster1.dcache-hit_latency`**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.dcache-maintenance_latency`**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.dcache-miss_latency`**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.dcache-prefetch_enabled`**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

Type: `bool`

Default value: `false`

#### **`subcluster1.dcache-read_access_latency`**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and

intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.dcache-read_latency`**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.dcache-snoop_data_transfer_latency`**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.dcache-write_access_latency`**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.dcache-write_latency`**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.force_zero_PSTATE_PAN`**

Non-architecture parameter to force `PSTATE.PAN` to be 0.0: No effect. 1: `PSTATE.PAN` is always treated as 0. The parameter optimizes the performance of updating `PSTATE.PAN`.



Type: `bool`

Default value: `false`

### **`subcluster1.has_delayed_dbgreg`**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

### **`subcluster1.has_delayed_sysreg`**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

### **`subcluster1.icache-hit_latency`**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.icache-maintenance_latency`**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.icache-miss_latency`**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.icache-prefetch_enabled`**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

### **`subcluster1.icache-read_access_latency`**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.icache-read_latency`**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.invalidate_code_cache_on_icache_cmo`**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: `uint8_t`

Default value: `0`

### **`subcluster1.ptw_latency`**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.tlb_latency`**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.tlbi_stall_enabled`**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: `bool`

Default value: `false`

**subcluster1.walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

3.58 ARMCortexA55CT\_CortexA76CT

Defined in `LISA/ARMCortexA55CT_CortexA76CT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
CortexA55 r1p0	Full support
CortexA76 r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

Changes in 11.31.15

The following parameters were added:

- `subcluster0.has_delayed_dbgreg`
- `subcluster0.has_delayed_sysreg`
- `subcluster1.has_delayed_dbgreg`
- `subcluster1.has_delayed_sysreg`

About ARMCortexA55CT\_CortexA76CT

The number of cores in each subcluster is configurable using the following parameters:

**subcluster0.NUM\_CORES**

Possible values are 1-7 (ARMCortexA55CT).

**subcluster1.NUM\_CORES**

Possible values are 1-4 (ARMCortexA76CT).

The total number of cores in the cluster cannot exceed 8.

Port arrays are expanded to the width needed for the maximum number of cores in each subcluster. Use the following port array indexes:

- `port_name[0-6]` for cores in subcluster0.

- `port_name[7-10]` for cores in subcluster1.

**Note**

All instances in the Master cross trigger matrix port array, `cti[11]` must be connected, regardless of the `NUM_CORES` value used.

Core-specific parameters have the following prefixes:

- `subcluster0.cpu0` to `subcluster0.cpu6` identify cores in `subcluster0`.
- `subcluster1.cpu0` to `subcluster1.cpu3` identify cores in `subcluster1`.

For information about the cores in this model, see:

- [ARMCortexA55CT](#)
- [ARMCortexA76CT](#)

### Iris and MTI instances for ARMCortexA55CT\_CortexA76CT

This model has the following Iris instances:

Name	Instance type
ARMCortexA55CT_CortexA76CT	Cluster_ARM_Cortex-A55_Cortex-A76
ARMCortexA55CT_CortexA76CT.AMU	PVBusLogger
ARMCortexA55CT_CortexA76CT.AMU.mapper	PVBusMapper
ARMCortexA55CT_CortexA76CT.DAP	PVBusLogger
ARMCortexA55CT_CortexA76CT.DAP.mapper	PVBusMapper
ARMCortexA55CT_CortexA76CT.DSU	DSU
ARMCortexA55CT_CortexA76CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA55CT_CortexA76CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA55CT_CortexA76CT.DSU.shared_cache	PVCache
ARMCortexA55CT_CortexA76CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA55CT_CortexA76CT.DSU.shared_cache.upstream[Z] (where Z = 0-5)	PVBusSlave
ARMCortexA55CT_CortexA76CT.MMAP	PVBusLogger
ARMCortexA55CT_CortexA76CT.MMAP.mapper	PVBusMapper
ARMCortexA55CT_CortexA76CT.RAS	PVBusLogger
ARMCortexA55CT_CortexA76CT.RAS.mapper	PVBusMapper
ARMCortexA55CT_CortexA76CT.ext_bus	PVBusLogger
ARMCortexA55CT_CortexA76CT.ext_bus.mapper	PVBusMapper
ARMCortexA55CT_CortexA76CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA55CT_CortexA76CT.global_debug_rom	debug_rom
ARMCortexA55CT_CortexA76CT.subcluster0	Subcluster_ARM_Cortex-A55
ARMCortexA55CT_CortexA76CT.subcluster0.cpu0	ARM_Cortex-A55
ARMCortexA55CT_CortexA76CT.subclusterZ.cpu0.UTLB (where Z = 0-1)	TLB

Name	Instance type
ARMCortexA55CT_CortexA76CT.subclusterZ.cpuU.dtlb (where Z = 0-1; U = 0-1)	TLB
ARMCortexA55CT_CortexA76CT.subclusterZ.cpu0.l1dcache (where Z = 0-1)	PVCache
ARMCortexA55CT_CortexA76CT.subclusterZ.cpu0.l1dcache.downstream[0].pvbusmaster (where Z = 0-1)	PVBusMaster
ARMCortexA55CT_CortexA76CT.subclusterZ.cpu0.l1dcache.upstream[0] (where Z = 0-1)	PVBusSlave
ARMCortexA55CT_CortexA76CT.subclusterZ.cpu0.l1icache (where Z = 0-1)	PVCache
ARMCortexA55CT_CortexA76CT.subclusterZ.cpu0.l1icache.downstream[0].pvbusmaster (where Z = 0-1)	PVBusMaster
ARMCortexA55CT_CortexA76CT.subclusterZ.cpu0.l1icache.upstream[0] (where Z = 0-1)	PVBusSlave
ARMCortexA55CT_CortexA76CT.subclusterZ.cpu0.l2cache (where Z = 0-1)	PVCache
ARMCortexA55CT_CortexA76CT.subclusterZ.cpu0.l2cache.downstream[0].pvbusmaster (where Z = 0-1)	PVBusMaster
ARMCortexA55CT_CortexA76CT.subclusterZ.cpu0.l2cache.upstream[W] (where Z = 0-1; W = 0-1)	PVBusSlave
ARMCortexA55CT_CortexA76CT.subcluster1	Subcluster_ARM_Cortex-A76
ARMCortexA55CT_CortexA76CT.subcluster1.cpu0	ARM_Cortex-A76

This model has the following MTI trace components:

Name	Component type
ARMCortexA55CT_CortexA76CT.AMU	PVBusLogger
ARMCortexA55CT_CortexA76CT.AMU.mapper	PVBusMapper
ARMCortexA55CT_CortexA76CT.DAP	PVBusLogger
ARMCortexA55CT_CortexA76CT.DAP.mapper	PVBusMapper
ARMCortexA55CT_CortexA76CT.DSU	DSU
ARMCortexA55CT_CortexA76CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA55CT_CortexA76CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA55CT_CortexA76CT.DSU.shared_cache	PVCache
ARMCortexA55CT_CortexA76CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA55CT_CortexA76CT.DSU.shared_cache.upstream[Z] (where Z = 0-5)	PVBusSlave
ARMCortexA55CT_CortexA76CT.MMAP	PVBusLogger
ARMCortexA55CT_CortexA76CT.MMAP.mapper	PVBusMapper
ARMCortexA55CT_CortexA76CT.RAS	PVBusLogger
ARMCortexA55CT_CortexA76CT.RAS.mapper	PVBusMapper
ARMCortexA55CT_CortexA76CT.ext_bus	PVBusLogger
ARMCortexA55CT_CortexA76CT.ext_bus.mapper	PVBusMapper
ARMCortexA55CT_CortexA76CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA55CT_CortexA76CT.subcluster0.cpu0	ARM_Cortex-A55
ARMCortexA55CT_CortexA76CT.subclusterZ.cpu0.UTLB (where Z = 0-1)	TLB
ARMCortexA55CT_CortexA76CT.subclusterZ.cpu0.l1dcache (where Z = 0-1)	PVCache

Name	Component type
ARMCortexA55CT_CortexA76CT.subclusterZ.cpu0.l1dcache.downstream[0].pvbusmaster (where Z = 0-1)	PVBusMaster
ARMCortexA55CT_CortexA76CT.subclusterZ.cpu0.l1dcache.upstream[0] (where Z = 0-1)	PVBusSlave
ARMCortexA55CT_CortexA76CT.subclusterZ.cpu0.l1icache (where Z = 0-1)	PVCache
ARMCortexA55CT_CortexA76CT.subclusterZ.cpu0.l1icache.downstream[0].pvbusmaster (where Z = 0-1)	PVBusMaster
ARMCortexA55CT_CortexA76CT.subclusterZ.cpu0.l1icache.upstream[0] (where Z = 0-1)	PVBusSlave
ARMCortexA55CT_CortexA76CT.subclusterZ.cpu0.l2cache (where Z = 0-1)	PVCache
ARMCortexA55CT_CortexA76CT.subclusterZ.cpu0.l2cache.downstream[0].pvbusmaster (where Z = 0-1)	PVBusMaster
ARMCortexA55CT_CortexA76CT.subclusterZ.cpu0.l2cache.upstream[W] (where Z = 0-1; W = 0-1)	PVBusSlave
ARMCortexA55CT_CortexA76CT.subcluster1.cpu0	ARM_Cortex-A76

### Ports for ARMCortexA55CT\_CortexA76CT

Port	Direction	Protocol	Description
aa64naa32	slave	Signal	Register width after reset.
acp_s	slave	PVBus	AXI ACP slave port
AENDMP	slave	Value_64	DynamiQ port to obtain end address of valid peripheral address range (exclusive).
ASTARTMP	slave	Value_64	DynamiQ port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	slave	Signal	CHI defined pins.
broadcastcachemaint	slave	Signal	ACE defined pins.
broadcastouter	slave	Signal	ACE defined pins.
broadcastpersist	slave	Signal	CHI defined pins.
cfgend	slave	Signal	This signal if for EE bit initialisation.
cfgsdisable	slave	Signal	This signal disables write access to some secure Interrupt Controller registers.
cfgte	slave	Signal	This signal provides default exception handling state.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
clusterid	slave	Value	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set ManagerID64 of the core.
clusterpmuirq	master	Signal	DynamiQ pmu irq
CNTHPIRQ	master	Signal	Timer signals to SOC.

Port	Direction	Protocol	Description
CNTHVIRQ	master	Signal	Timer signals to SOC.
CNTPNSIRQ	master	Signal	Timer signals to SOC.
CNTPSIRQ	master	Signal	Timer signals to SOC.
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.
CNTVIRQ	master	Signal	Timer signals to SOC.
commirq	master	Signal	Interrupt signal from debug communications channel.
core_clk_in	slave	ClockSignal	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
cp15sdisable	slave	Signal	This signal disables write access to some system control processor registers.
cpuporeset	slave	Signal	CPU power on reset. Initializes all the processor logic, including debug logic.
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
ctidbgirq	master	Signal	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	slave	Signal	External debug interface.
dbgnopwrdown	master	Signal	No power-down request.
dbgpwrupreq	master	Signal	Debug power up request.
dev_debug_s	slave	PVBus	External debug interface.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	slave	PVBus	External debug interface.
niden	slave	Signal	External debug interface.
pchannel_cluster	slave	PChannel	PChannel for cluster.
pchannel_core	slave	PChannel	PChannels for cores
pmbirq	master	Signal	Interrupt signal from statistical profiling unit.
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	master	PVBus	The core will generate bus requests on this port.
pvbus_periph_m	master	PVBus	The core can generate peripheral bus request on this port.
rei	slave	Signal	Per core RAM Error Interrupt.
reset	slave	Signal	Raising this signal will put the core into reset mode.
rvbaraddr	slave	Value_64	Reset vector base address.
sci_m	master	SystemCoherencyInterface	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei	slave	Signal	Per core System Error physical pins.

Port	Direction	Protocol	Description
spiden	slave	Signal	External debug interface.
spniden	slave	Signal	External debug interface.
sporeset	slave	Signal	A single cluster-wide power on reset signal for all resettable registers in DynamIQ.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfiq	slave	Signal	Virtualised FIQ.
vinithi	slave	Signal	This signal controls of the location of the exception vectors at reset.
virq	slave	Signal	Virtualised IRQ.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L3 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Per core virtual System Error physical pins.

## Parameters for ARM CortexA55CT\_CortexA76CT

### BROADCASTATOMIC

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`

Default value: `true`

### BROADCASTCACHEMAINT

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

Type: `bool`

Default value: `true`

### BROADCASTOUTER

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

Type: `bool`

Default value: `true`

### BROADCASTPERSIST

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.



Type: `bool`

Default value: `true`

### **CLUSTER\_ID**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the ManagerID64 of the core.

Type: `uint32_t`

Default value: `0x0`

### **CMO\_broadcast\_when\_cache\_state\_modelling\_disabled**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. This value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required.

Type: `uint8_t`

Default value: `1`

### **GICDISABLE**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`

Default value: `true`

### **bus\_type**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

Type: `uint8_t`

Default value: `0`

### **dcache-state\_modelled**

Set whether D-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **default\_opmode**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

Type: `uint8_t`

Default value: 4

### **diagnostics**

Enable DynamIQ diagnostic messages.

Type: `bool`

Default value: `false`

### **enable\_simulation\_performance\_optimizations**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of `stage12_tlb_size` parameter to 1024).

Type: `bool`

Default value: `true`

### **has\_acp**

If true, Accelerator Coherency Port is configured.

Type: `bool`

Default value: `false`

### **has\_peripheral\_port**

If true, additional AXI peripheral port is configured.

Type: `bool`

Default value: `false`

### **icache-state\_modelled**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

**l3cache-hit\_latency**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-maintenance\_latency**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-miss\_latency**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-read\_access\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size `l3cache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-read\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l3cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-size**

L3 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

### **`l3cache-snoop_data_transfer_latency`**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-snoop_issue_latency`**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-write_access_latency`**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size `l3cache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-write_latency`**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l3cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`pchannel_treat_simreset_as_poreset`**

Register core as ON state to cluster with simulation reset.

Type: `bool`

Default value: `false`

### **`periph_address_end`**

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

Type: `uint64_t`

Default value: `0x0`

### **`periph_address_start`**

Start address for peripheral port address range inclusive (corresponds to ASTARTMP input signal).

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.NUM_CORES`**

Number of cores per cluster.

Type: `uint8_t`

Default value: `1`

### **`subcluster0.cpi_div`**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

### **`subcluster0.cpi_mul`**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

### **`subcluster0.cpu0.AA64nAA32`**

Register width configuration at reset. 0, AArch32. 1, AArch64.

Type: `bool`

Default value: `true`

### **`subcluster0.cpu0.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster0.cpu0.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster0.cpu0.CP15SDISABLE**

Initialize to disable access to some CP15 registers.

Type: `bool`

Default value: `false`

**subcluster0.cpu0.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster0.cpu0.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu0.VINITHI**

Reset value of SCTLR.V.

Type: `bool`

Default value: `false`

**subcluster0.cpu0.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu0.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

#### **subcluster0.cpu0.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu0.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu0.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu0.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu0.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x40000

#### **subcluster0.cpu0.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu0.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu0.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu0.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu0.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

#### **`subcluster0.cpu0.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

#### **`subcluster0.cpu0.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.



Type: `uint16_t`

Default value: `0xf000`

**`subcluster0.cpu0.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**`subcluster0.cpu0.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**`subcluster0.cpu0.semihosting-T32_HLT`**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**`subcluster0.cpu0.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**`subcluster0.cpu0.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**`subcluster0.cpu0.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**`subcluster0.cpu0.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster0.cpu0.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu0.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu0.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster0.cpu0.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu0.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu0.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster0.cpu0.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster0.cpu1.AA64nAA32**

Register width configuration at reset. 0, AArch32. 1, AArch64.

Type: `bool`

Default value: `true`

**subcluster0.cpu1.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster0.cpu1.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster0.cpu1.CP15SDISABLE**

Initialize to disable access to some CP15 registers.

Type: `bool`

Default value: `false`

**subcluster0.cpu1.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster0.cpu1.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu1.VINITHI`**

Reset value of SCTLR.V.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu1.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu1.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu1.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu1.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu1.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu1.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu1.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x40000`

### **`subcluster0.cpu1.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu1.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu1.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu1.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu1.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu1.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu1.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu1.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu1.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

### **`subcluster0.cpu1.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

### **`subcluster0.cpu1.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **`subcluster0.cpu1.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **`subcluster0.cpu1.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster0.cpu1.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu1.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu1.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster0.cpu1.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu1.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu1.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu1.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu2.AA64nAA32`**

Register width configuration at reset. 0, AArch32. 1, AArch64.

Type: `bool`

Default value: `true`

### **`subcluster0.cpu2.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`



Default value: `false`

**subcluster0.cpu2.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster0.cpu2.CP15SDISABLE**

Initialize to disable access to some CP15 registers.

Type: `bool`

Default value: `false`

**subcluster0.cpu2.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster0.cpu2.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu2.VINITHI**

Reset value of SCTLR.V.

Type: `bool`

Default value: `false`

**subcluster0.cpu2.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu2.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu2.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu2.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu2.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu2.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu2.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x40000`

**subcluster0.cpu2.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu2.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu2.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu2.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu2.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu2.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster0.cpu2.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster0.cpu2.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu2.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu2.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu2.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu2.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu2.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu2.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu2.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu2.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu2.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu3.AA64nAA32`**

Register width configuration at reset. 0, AArch32. 1, AArch64.

Type: `bool`

Default value: `true`

### **`subcluster0.cpu3.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu3.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu3.CP15SDISABLE`**

Initialize to disable access to some CP15 registers.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu3.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster0.cpu3.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu3.VINITHI**

Reset value of SCTLR.V.

Type: `bool`

Default value: `false`

**subcluster0.cpu3.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu3.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu3.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu3.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu3.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x40000

**subcluster0.cpu3.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.



Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu3.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu3.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

### **`subcluster0.cpu3.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

### **`subcluster0.cpu3.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu3.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu3.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu3.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster0.cpu3.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster0.cpu3.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster0.cpu3.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster0.cpu3.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu3.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu3.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu3.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu3.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu3.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu3.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster0.cpu3.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster0.cpu4.AA64nAA32**

Register width configuration at reset. 0, AArch32. 1, AArch64.

Type: `bool`

Default value: `true`

**subcluster0.cpu4.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster0.cpu4.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster0.cpu4.CP15SDISABLE**

Initialize to disable access to some CP15 registers.

Type: `bool`

Default value: `false`

**subcluster0.cpu4.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster0.cpu4.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu4.VINITHI**

Reset value of SCTLR.V.

Type: `bool`

Default value: `false`

**subcluster0.cpu4.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu4.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu4.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu4.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu4.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu4.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu4.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x40000`

#### **`subcluster0.cpu4.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu4.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu4.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu4.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu4.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

#### **`subcluster0.cpu4.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: 0

#### **`subcluster0.cpu4.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

#### **`subcluster0.cpu4.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

#### **`subcluster0.cpu4.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: 0x123456

#### **`subcluster0.cpu4.semihosting-T32_HLT`**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

#### **`subcluster0.cpu4.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

#### **`subcluster0.cpu4.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

#### **`subcluster0.cpu4.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

### **`subcluster0.cpu4.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster0.cpu4.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu4.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu4.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster0.cpu4.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu4.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`



**subcluster0.cpu4.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster0.cpu4.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster0.cpu5.AA64nAA32**

Register width configuration at reset. 0, AArch32. 1, AArch64.

Type: `bool`

Default value: `true`

**subcluster0.cpu5.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster0.cpu5.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster0.cpu5.CP15SDISABLE**

Initialize to disable access to some CP15 registers.

Type: `bool`

Default value: `false`

**subcluster0.cpu5.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu5.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu5.VINITHI`**

Reset value of SCTLR.V.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu5.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu5.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu5.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu5.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu5.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x40000

**subcluster0.cpu5.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu5.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu5.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

### **`subcluster0.cpu5.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

### **`subcluster0.cpu5.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu5.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu5.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu5.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster0.cpu5.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster0.cpu5.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster0.cpu5.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster0.cpu5.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu5.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: 0x0

**subcluster0.cpu5.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: 0xf000000

**subcluster0.cpu5.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu5.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu5.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu5.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster0.cpu5.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster0.cpu6.AA64nAA32**

Register width configuration at reset. 0, AArch32. 1, AArch64.

Type: `bool`

Default value: `true`

**subcluster0.cpu6.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster0.cpu6.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster0.cpu6.CP15SDISABLE**

Initialize to disable access to some CP15 registers.

Type: `bool`

Default value: `false`

**subcluster0.cpu6.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster0.cpu6.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu6.VINITHI**

Reset value of SCTLR.V.

Type: `bool`

Default value: `false`

**subcluster0.cpu6.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu6.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu6.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu6.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu6.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu6.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu6.l2cache-size`**

L2 Cache size in bytes.



Type: `uint32_t`

Default value: `0x40000`

### **`subcluster0.cpu6.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu6.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu6.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu6.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu6.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

### **`subcluster0.cpu6.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: 0

#### **`subcluster0.cpu6.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

#### **`subcluster0.cpu6.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

#### **`subcluster0.cpu6.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: 0x123456

#### **`subcluster0.cpu6.semihosting-T32_HLT`**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

#### **`subcluster0.cpu6.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

#### **`subcluster0.cpu6.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

#### **`subcluster0.cpu6.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

### **`subcluster0.cpu6.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster0.cpu6.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu6.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu6.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster0.cpu6.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu6.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu6.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster0.cpu6.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster0.dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.dcache-maintenance\_latency**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.dcache-miss\_latency**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.dcache-prefetch\_enabled**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

Type: `bool`

Default value: `false`

**subcluster0.dcache-read\_access\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.dcache-size**

L1 D-Cache size in bytes.

Type: uint32\_t

Default value: 0x8000

**subcluster0.dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.dcache-write\_access\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.force\_zero\_PSTATE\_PAN**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: bool

Default value: false

**subcluster0.has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: bool

Default value: false

**subcluster0.has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: bool

Default value: true

**subcluster0.has\_dot\_product**

Possible values of this parameter are:- 1, feature not implemented.- 2, feature is implemented.

Type: uint8\_t

Default value: 2

**subcluster0.icache-hit\_latency**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

**subcluster0.icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.icache-size**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: `0x8000`

#### **`subcluster0.invalidate_code_cache_on_icache_cmo`**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: `uint8_t`

Default value: `0`

#### **`subcluster0.ptw_latency`**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.tlb_latency`**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.tlbi_stall_enabled`**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: `bool`

Default value: `false`

#### **`subcluster0.walk_cache_latency`**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.NUM_CORES`**

Number of cores per cluster.

Type: `uint8_t`

Default value: `1`



**subcluster1.cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**subcluster1.cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**subcluster1.cpu0.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu0.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster1.cpu0.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu0.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu0.VINITHI**

Reset value of SCTLR.V.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu0.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu0.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu0.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu0.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu0.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu0.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu0.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

### **`subcluster1.cpu0.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu0.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu0.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu0.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu0.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

#### **`subcluster1.cpu0.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

#### **`subcluster1.cpu0.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster1.cpu0.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster1.cpu0.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

#### **`subcluster1.cpu0.semihosting-T32_HLT`**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

#### **`subcluster1.cpu0.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

#### **`subcluster1.cpu0.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**`subcluster1.cpu0.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**`subcluster1.cpu0.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**`subcluster1.cpu0.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**`subcluster1.cpu0.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**`subcluster1.cpu0.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**`subcluster1.cpu0.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu0.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu0.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster1.cpu1.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: false

**subcluster1.cpu1.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: bool

Default value: false

**subcluster1.cpu1.CRYPTODISABLE**

Disable cryptographic features.

Type: bool

Default value: false

**subcluster1.cpu1.RVBARADDR**

Value of RVBAR\_ELx register.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu1.VINITHI**

Reset value of SCTLR.V.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu1.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu1.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to

the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu1.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

#### **`subcluster1.cpu1.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu1.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu1.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu1.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`



**subcluster1.cpu1.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu1.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu1.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu1.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu1.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu1.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster1.cpu1.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster1.cpu1.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu1.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu1.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu1.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu1.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu1.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu1.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: 0xf000000

**subcluster1.cpu1.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu1.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster1.cpu2.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: false

**subcluster1.cpu2.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: bool

Default value: false

**subcluster1.cpu2.CRYPTODISABLE**

Disable cryptographic features.

Type: bool

Default value: false

**subcluster1.cpu2.RVBARADDR**

Value of RVBAR\_ELx register.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu2.VINITHI**

Reset value of SCTL.R.V.

Type: `bool`

Default value: `false`

**subcluster1.cpu2.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu2.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

**subcluster1.cpu2.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster1.cpu2.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu2.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu2.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu2.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu2.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu2.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster1.cpu2.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

### **`subcluster1.cpu2.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **`subcluster1.cpu2.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **`subcluster1.cpu2.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster1.cpu2.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu2.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster1.cpu2.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster1.cpu2.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

#### **`subcluster1.cpu2.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster1.cpu2.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu3.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu3.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu3.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu3.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`



**subcluster1.cpu3.VINITHI**

Reset value of SCTL.R.V.

Type: `bool`

Default value: `false`

**subcluster1.cpu3.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu3.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster1.cpu3.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu3.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu3.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu3.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu3.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu3.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster1.cpu3.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

### **`subcluster1.cpu3.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **`subcluster1.cpu3.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **`subcluster1.cpu3.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster1.cpu3.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu3.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster1.cpu3.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster1.cpu3.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster1.cpu3.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu3.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster1.dcache-hit_latency`**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.dcache-maintenance_latency`**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.dcache-miss_latency`**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.dcache-prefetch_enabled`**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

Type: `bool`

Default value: `false`

### **`subcluster1.dcache-read_access_latency`**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.dcache-read_latency`**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.dcache-snoop_data_transfer_latency`**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.dcache-write_access_latency`**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.dcache-write_latency`**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster1.ext\_abort\_device\_read\_is\_sync**

Synchronous reporting of device-nGnRE read external aborts.

Type: bool

Default value: false

**subcluster1.ext\_abort\_device\_write\_is\_sync**

Synchronous reporting of device-nGnRE write external aborts.

Type: bool

Default value: false

**subcluster1.ext\_abort\_so\_read\_is\_sync**

Synchronous reporting of device-nGnRnE read external aborts.

Type: bool

Default value: false

**subcluster1.ext\_abort\_so\_write\_is\_sync**

Synchronous reporting of device-nGnRnE write external aborts.

Type: bool

Default value: false

**subcluster1.force\_zero\_PSTATE\_PAN**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: bool

Default value: false

**subcluster1.has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: bool

Default value: false

**subcluster1.has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

### **`subcluster1.has_statistical_profiling`**

Whether Statistical Based Profiling is implemented (FEAT\_SPE).

Type: `bool`

Default value: `true`

### **`subcluster1.icache-hit_latency`**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.icache-maintenance_latency`**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.icache-miss_latency`**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.icache-prefetch_enabled`**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

### **`subcluster1.icache-read_access_latency`**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and



intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.icache-read_latency`**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.invalidate_code_cache_on_icache_cmo`**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: `uint8_t`

Default value: `0`

#### **`subcluster1.ptw_latency`**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.tlb_latency`**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.tlbi_stall_enabled`**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: `bool`

Default value: `false`

**subcluster1.treat-dcache-cmos-to-pou-as-nop**

Whether dcache invalidation to the point of unification is required for instruction to data coherence. 0 - Invalidate ops required, 1 - Invalidate ops not required and cannot generate faults, 2 - Invalidate ops not required but can generate faults.

Type: uint8\_t

Default value: 0

**subcluster1.walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

### 3.59 ARMCortexA55CT\_CortexA78CT

Defined in LISA/ARMCortexA55CT\_CortexA78CT.lisa.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
CortexA55 r1p0	Full support
CortexA78 r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

**Changes in 11.31.15**

The following parameters were added:

- subcluster0.has\_delayed\_dbgreg
- subcluster0.has\_delayed\_sysreg
- subcluster1.has\_delayed\_dbgreg
- subcluster1.has\_delayed\_sysreg

**About ARMCortexA55CT\_CortexA78CT**

The number of cores in each subcluster is configurable using the following parameters:

**subcluster0.NUM\_CORES**

Possible values are 1-7 (ARMCortexA55CT).

**subcluster1.NUM\_CORES**

Possible values are 1-4 (ARMCortexA78CT).

The total number of cores in the cluster cannot exceed 8.

Port arrays are expanded to the width needed for the maximum number of cores in each subcluster. Use the following port array indexes:

- port\_name[0-6] for cores in subcluster0.
- port\_name[7-10] for cores in subcluster1.



All instances in the Master cross trigger matrix port array, `cti[11]` must be connected, regardless of the `NUM_CORES` value used.

Core-specific parameters have the following prefixes:

- subcluster0.cpu0 to subcluster0.cpu6 identify cores in subcluster0.
- subcluster1.cpu0 to subcluster1.cpu3 identify cores in subcluster1.

For information about the cores in this model, see:

- [ARMCortexA55CT](#)
- [ARMCortexA78CT](#)

## Iris and MTI instances for ARMCortexA55CT\_CortexA78CT

This model has the following Iris instances:

Name	Instance type
ARMCortexA55CT_CortexA78CT	Cluster_ARM_Cortex-A55_Cortex-A78
ARMCortexA55CT_CortexA78CT.AMU	PVBusLogger
ARMCortexA55CT_CortexA78CT.AMU.mapper	PVBusMapper
ARMCortexA55CT_CortexA78CT.DAP	PVBusLogger
ARMCortexA55CT_CortexA78CT.DAP.mapper	PVBusMapper
ARMCortexA55CT_CortexA78CT.DSU	DSU
ARMCortexA55CT_CortexA78CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA55CT_CortexA78CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA55CT_CortexA78CT.DSU.shared_cache	PVCache
ARMCortexA55CT_CortexA78CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA55CT_CortexA78CT.DSU.shared_cache.upstream[Z] (where Z = 0-5)	PVBusSlave
ARMCortexA55CT_CortexA78CT.MMAP	PVBusLogger
ARMCortexA55CT_CortexA78CT.MMAP.mapper	PVBusMapper
ARMCortexA55CT_CortexA78CT.RAS	PVBusLogger
ARMCortexA55CT_CortexA78CT.RAS.mapper	PVBusMapper
ARMCortexA55CT_CortexA78CT.ext_bus	PVBusLogger
ARMCortexA55CT_CortexA78CT.ext_bus.mapper	PVBusMapper
ARMCortexA55CT_CortexA78CT.gic_cpuif_decoder_cluster	GLCv3CPUInterfaceDecoder
ARMCortexA55CT_CortexA78CT.global_debug_rom	debug_rom

Name	Instance type
ARMCortexA55CT_CortexA78CT.subcluster0	Subcluster_ARM_Cortex-A55
ARMCortexA55CT_CortexA78CT.subcluster0.cpu0	ARM_Cortex-A55
ARMCortexA55CT_CortexA78CT.subclusterZ.cpu0.UTLB (where Z = 0-1)	TLB
ARMCortexA55CT_CortexA78CT.subclusterZ.cpuU.dtlb (where Z = 0-1; U = 0-1)	TLB
ARMCortexA55CT_CortexA78CT.subclusterZ.cpu0.l1dcache (where Z = 0-1)	PVCache
ARMCortexA55CT_CortexA78CT.subclusterZ.cpu0.l1dcache.downstream[0].pvbusmaster (where Z = 0-1)	PVBusMaster
ARMCortexA55CT_CortexA78CT.subclusterZ.cpu0.l1dcache.upstream[0] (where Z = 0-1)	PVBusSlave
ARMCortexA55CT_CortexA78CT.subclusterZ.cpu0.l1icache (where Z = 0-1)	PVCache
ARMCortexA55CT_CortexA78CT.subclusterZ.cpu0.l1icache.downstream[0].pvbusmaster (where Z = 0-1)	PVBusMaster
ARMCortexA55CT_CortexA78CT.subclusterZ.cpu0.l1icache.upstream[0] (where Z = 0-1)	PVBusSlave
ARMCortexA55CT_CortexA78CT.subclusterZ.cpu0.l2cache (where Z = 0-1)	PVCache
ARMCortexA55CT_CortexA78CT.subclusterZ.cpu0.l2cache.downstream[0].pvbusmaster (where Z = 0-1)	PVBusMaster
ARMCortexA55CT_CortexA78CT.subclusterZ.cpu0.l2cache.upstream[W] (where Z = 0-1; W = 0-1)	PVBusSlave
ARMCortexA55CT_CortexA78CT.subcluster1	Subcluster_ARM_Cortex-A78
ARMCortexA55CT_CortexA78CT.subcluster1.cpu0	ARM_Cortex-A78

This model has the following MTI trace components:

Name	Component type
ARMCortexA55CT_CortexA78CT.AMU	PVBusLogger
ARMCortexA55CT_CortexA78CT.AMU.mapper	PVBusMapper
ARMCortexA55CT_CortexA78CT.DAP	PVBusLogger
ARMCortexA55CT_CortexA78CT.DAP.mapper	PVBusMapper
ARMCortexA55CT_CortexA78CT.DSU	DSU
ARMCortexA55CT_CortexA78CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA55CT_CortexA78CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA55CT_CortexA78CT.DSU.shared_cache	PVCache
ARMCortexA55CT_CortexA78CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA55CT_CortexA78CT.DSU.shared_cache.upstream[Z] (where Z = 0-5)	PVBusSlave
ARMCortexA55CT_CortexA78CT.MMAP	PVBusLogger
ARMCortexA55CT_CortexA78CT.MMAP.mapper	PVBusMapper
ARMCortexA55CT_CortexA78CT.RAS	PVBusLogger
ARMCortexA55CT_CortexA78CT.RAS.mapper	PVBusMapper
ARMCortexA55CT_CortexA78CT.ext_bus	PVBusLogger
ARMCortexA55CT_CortexA78CT.ext_bus.mapper	PVBusMapper
ARMCortexA55CT_CortexA78CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA55CT_CortexA78CT.subcluster0.cpu0	ARM_Cortex-A55

Name	Component type
ARMCortexA55CT_CortexA78CT.subclusterZ.cpu0.UTLB (where Z = 0-1)	TLB
ARMCortexA55CT_CortexA78CT.subclusterZ.cpu0.l1dcache (where Z = 0-1)	PVCache
ARMCortexA55CT_CortexA78CT.subclusterZ.cpu0.l1dcache.downstream[0].pvbusmaster (where Z = 0-1)	PVBusMaster
ARMCortexA55CT_CortexA78CT.subclusterZ.cpu0.l1dcache.upstream[0] (where Z = 0-1)	PVBusSlave
ARMCortexA55CT_CortexA78CT.subclusterZ.cpu0.l1icache (where Z = 0-1)	PVCache
ARMCortexA55CT_CortexA78CT.subclusterZ.cpu0.l1icache.downstream[0].pvbusmaster (where Z = 0-1)	PVBusMaster
ARMCortexA55CT_CortexA78CT.subclusterZ.cpu0.l1icache.upstream[0] (where Z = 0-1)	PVBusSlave
ARMCortexA55CT_CortexA78CT.subclusterZ.cpu0.l2cache (where Z = 0-1)	PVCache
ARMCortexA55CT_CortexA78CT.subclusterZ.cpu0.l2cache.downstream[0].pvbusmaster (where Z = 0-1)	PVBusMaster
ARMCortexA55CT_CortexA78CT.subclusterZ.cpu0.l2cache.upstream[W] (where Z = 0-1; W = 0-1)	PVBusSlave
ARMCortexA55CT_CortexA78CT.subcluster1.cpu0	ARM_Cortex-A78

### Ports for ARMCortexA55CT\_CortexA78CT

Port	Direction	Protocol	Description
aa64naa32	slave	Signal	Register width after reset.
acp_s	slave	PVBus	AXI ACP slave port
AENDMP	slave	Value_64	DynamiQ port to obtain end address of valid peripheral address range (exclusive).
ASTARTMP	slave	Value_64	DynamiQ port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	slave	Signal	CHI defined pins.
broadcastcachemaint	slave	Signal	ACE defined pins.
broadcastouter	slave	Signal	ACE defined pins.
broadcastpersist	slave	Signal	CHI defined pins.
cfgend	slave	Signal	This signal if for EE bit initialisation.
cfgsdisable	slave	Signal	This signal disables write access to some secure Interrupt Controller registers.
cfgte	slave	Signal	This signal provides default exception handling state.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
clusterid	slave	Value	The port reads the value in CPU ID register field, bits[11:14] of the MPIDR.
clusterpmuirq	master	Signal	DynamiQ pmu irq
CNTHPIRQ	master	Signal	Timer signals to SOC.
CNTHVIRQ	master	Signal	Timer signals to SOC.
CNTPNSIRQ	master	Signal	Timer signals to SOC.
CNTPSIRQ	master	Signal	Timer signals to SOC.

Port	Direction	Protocol	Description
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.
CNTVIRQ	master	Signal	Timer signals to SOC.
commirq	master	Signal	Interrupt signal from debug communications channel.
core_clk_in	slave	ClockSignal	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
cp15sdisable	slave	Signal	This signal disables write access to some system control processor registers.
cpuporeset	slave	Signal	CPU power on reset. Initializes all the processor logic, including debug logic.
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
ctidbgirq	master	Signal	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	slave	Signal	External debug interface.
dbgnopwrdown	master	Signal	No power-down request.
dbgprupreq	master	Signal	Debug power up request.
dev_debug_s	slave	PVBus	External debug interface.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	slave	PVBus	External debug interface.
niden	slave	Signal	External debug interface.
pchannel_cluster	slave	PChannel	PChannel for cluster.
pchannel_core	slave	PChannel	PChannels for cores
pmbirq	master	Signal	Interrupt signal from statistical profiling unit.
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	master	PVBus	The core will generate bus requests on this port.
pvbus_periph_m	master	PVBus	The core can generate peripheral bus request on this port.
rei	slave	Signal	Per core RAM Error Interrupt.
reset	slave	Signal	Raising this signal will put the core into reset mode.
rvbaraddr	slave	Value_64	Reset vector base address.
sci_m	master	SystemCoherencyInterface	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei	slave	Signal	Per core System Error physical pins.
spiden	slave	Signal	External debug interface.
spniden	slave	Signal	External debug interface.

Port	Direction	Protocol	Description
sporeset	slave	Signal	A single cluster-wide power on reset signal for all resettable registers in DynamIQ.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfiq	slave	Signal	Virtualised FIQ.
vinithi	slave	Signal	This signal controls of the location of the exception vectors at reset.
virq	slave	Signal	Virtualised IRQ.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L3 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Per core virtual System Error physical pins.

## Parameters for ARM CortexA55CT\_CortexA78CT

### BROADCASTATOMIC

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`

Default value: `true`

### BROADCASTCACHEMAINT

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

Type: `bool`

Default value: `true`

### BROADCASTOUTER

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

Type: `bool`

Default value: `true`

### BROADCASTPERSIST

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: `bool`

Default value: `true`

### **CLUSTER\_ID**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the ManagerID64 of the core.

Type: `uint32_t`

Default value: `0x0`

### **CMO\_broadcast\_when\_cache\_state\_modelling\_disabled**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. This value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required.

Type: `uint8_t`

Default value: `1`

### **GICDISABLE**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`

Default value: `true`

### **bus\_type**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

Type: `uint8_t`

Default value: `0`

### **dcache-state\_modelled**

Set whether D-cache has stateful implementation.

Type: `bool`

Default value: `false`



**default\_opmode**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

Type: `uint8_t`

Default value: 4

**diagnostics**

Enable DynamIQ diagnostic messages.

Type: `bool`

Default value: `false`

**enable\_simulation\_performance\_optimizations**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of `stage12_tlb_size` parameter to 1024).

Type: `bool`

Default value: `true`

**has\_acp**

If true, Accelerator Coherency Port is configured.

Type: `bool`

Default value: `false`

**has\_peripheral\_port**

If true, additional AXI peripheral port is configured.

Type: `bool`

Default value: `false`

**icache-state\_modelled**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

**l3cache-hit\_latency**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-maintenance\_latency**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-miss\_latency**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-read\_access\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size `l3cache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-read\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l3cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-size**

L3 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

### **`l3cache-snoop_data_transfer_latency`**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-snoop_issue_latency`**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-write_access_latency`**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size `l3cache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-write_latency`**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l3cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`pchannel_treat_simreset_as_poreset`**

Register core as ON state to cluster with simulation reset.

Type: `bool`

Default value: `false`

### **`periph_address_end`**

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

Type: `uint64_t`

Default value: `0x0`

### **`periph_address_start`**

Start address for peripheral port address range inclusive (corresponds to ASTARTMP input signal).

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.NUM_CORES`**

Number of cores per cluster.

Type: `uint8_t`

Default value: `1`

### **`subcluster0.cpi_div`**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

### **`subcluster0.cpi_mul`**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

### **`subcluster0.cpu0.AA64nAA32`**

Register width configuration at reset. 0, AArch32. 1, AArch64.

Type: `bool`

Default value: `true`

### **`subcluster0.cpu0.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster0.cpu0.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster0.cpu0.CP15SDISABLE**

Initialize to disable access to some CP15 registers.

Type: `bool`

Default value: `false`

**subcluster0.cpu0.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster0.cpu0.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu0.VINITHI**

Reset value of SCTLR.V.

Type: `bool`

Default value: `false`

**subcluster0.cpu0.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu0.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

#### **subcluster0.cpu0.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu0.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu0.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu0.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu0.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x40000

#### **subcluster0.cpu0.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu0.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu0.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu0.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu0.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

#### **`subcluster0.cpu0.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

#### **`subcluster0.cpu0.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**`subcluster0.cpu0.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**`subcluster0.cpu0.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**`subcluster0.cpu0.semihosting-T32_HLT`**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**`subcluster0.cpu0.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**`subcluster0.cpu0.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**`subcluster0.cpu0.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**`subcluster0.cpu0.semihosting-enable`**

Enable semihosting SVC/HLT traps.



Type: `bool`

Default value: `true`

### **`subcluster0.cpu0.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu0.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu0.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster0.cpu0.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu0.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu0.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster0.cpu0.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster0.cpu1.AA64nAA32**

Register width configuration at reset. 0, AArch32. 1, AArch64.

Type: `bool`

Default value: `true`

**subcluster0.cpu1.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster0.cpu1.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster0.cpu1.CP15SDISABLE**

Initialize to disable access to some CP15 registers.

Type: `bool`

Default value: `false`

**subcluster0.cpu1.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster0.cpu1.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu1.VINITHI`**

Reset value of SCTLR.V.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu1.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu1.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu1.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu1.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu1.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu1.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu1.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x40000`

### **`subcluster0.cpu1.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu1.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu1.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu1.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu1.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu1.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu1.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu1.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu1.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**`subcluster0.cpu1.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**`subcluster0.cpu1.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**`subcluster0.cpu1.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**`subcluster0.cpu1.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**`subcluster0.cpu1.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**`subcluster0.cpu1.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**`subcluster0.cpu1.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster0.cpu1.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu1.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu1.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu1.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu2.AA64nAA32`**

Register width configuration at reset. 0, AArch32. 1, AArch64.

Type: `bool`

Default value: `true`

### **`subcluster0.cpu2.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu2.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu2.CP15SDISABLE`**

Initialize to disable access to some CP15 registers.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu2.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu2.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu2.VINITHI`**

Reset value of SCTLR.V.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu2.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu2.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.



Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu2.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu2.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu2.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu2.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu2.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x40000`

**subcluster0.cpu2.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu2.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu2.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu2.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu2.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu2.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu2.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster0.cpu2.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster0.cpu2.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu2.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu2.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu2.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu2.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu2.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu2.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu2.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu2.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu2.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu3.AA64nAA32`**

Register width configuration at reset. 0, AArch32. 1, AArch64.

Type: `bool`

Default value: `true`

### **`subcluster0.cpu3.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu3.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu3.CP15SDISABLE`**

Initialize to disable access to some CP15 registers.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu3.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster0.cpu3.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu3.VINITHI**

Reset value of SCTLR.V.

Type: `bool`

Default value: `false`

**subcluster0.cpu3.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu3.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu3.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu3.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu3.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x40000

**subcluster0.cpu3.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu3.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu3.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

### **`subcluster0.cpu3.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

### **`subcluster0.cpu3.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu3.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu3.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`



**subcluster0.cpu3.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster0.cpu3.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster0.cpu3.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster0.cpu3.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster0.cpu3.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu3.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu3.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu3.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu3.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu3.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu3.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster0.cpu3.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster0.cpu4.AA64nAA32**

Register width configuration at reset. 0, AArch32. 1, AArch64.

Type: `bool`

Default value: `true`

**subcluster0.cpu4.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster0.cpu4.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster0.cpu4.CP15SDISABLE**

Initialize to disable access to some CP15 registers.

Type: `bool`

Default value: `false`

**subcluster0.cpu4.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster0.cpu4.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu4.VINITHI**

Reset value of SCTLR.V.

Type: `bool`

Default value: `false`

**subcluster0.cpu4.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu4.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu4.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu4.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu4.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu4.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu4.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x40000`

#### **`subcluster0.cpu4.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu4.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu4.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu4.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu4.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

#### **`subcluster0.cpu4.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: 0

**`subcluster0.cpu4.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**`subcluster0.cpu4.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**`subcluster0.cpu4.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: 0x123456

**`subcluster0.cpu4.semihosting-T32_HLT`**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**`subcluster0.cpu4.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**`subcluster0.cpu4.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**`subcluster0.cpu4.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

### **`subcluster0.cpu4.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster0.cpu4.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu4.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu4.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster0.cpu4.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu4.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu4.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster0.cpu4.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster0.cpu5.AA64nAA32**

Register width configuration at reset. 0, AArch32. 1, AArch64.

Type: `bool`

Default value: `true`

**subcluster0.cpu5.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster0.cpu5.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster0.cpu5.CP15SDISABLE**

Initialize to disable access to some CP15 registers.

Type: `bool`

Default value: `false`

**subcluster0.cpu5.CRYPTODISABLE**

Disable cryptographic features.



Type: `bool`

Default value: `false`

### **`subcluster0.cpu5.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu5.VINITHI`**

Reset value of SCTLR.V.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu5.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu5.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu5.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu5.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu5.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x40000

**subcluster0.cpu5.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu5.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu5.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

### **`subcluster0.cpu5.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

### **`subcluster0.cpu5.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu5.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu5.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu5.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster0.cpu5.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster0.cpu5.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster0.cpu5.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster0.cpu5.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu5.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu5.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu5.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu5.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu5.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu5.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster0.cpu5.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster0.cpu6.AA64nAA32**

Register width configuration at reset. 0, AArch32. 1, AArch64.

Type: `bool`

Default value: `true`

**subcluster0.cpu6.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster0.cpu6.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster0.cpu6.CP15SDISABLE**

Initialize to disable access to some CP15 registers.

Type: `bool`

Default value: `false`

**subcluster0.cpu6.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster0.cpu6.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu6.VINITHI**

Reset value of SCTLR.V.

Type: `bool`

Default value: `false`

**subcluster0.cpu6.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu6.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu6.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu6.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu6.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu6.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu6.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x40000`

### **`subcluster0.cpu6.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu6.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu6.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu6.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu6.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

### **`subcluster0.cpu6.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).



Type: `uint8_t`

Default value: 0

#### **`subcluster0.cpu6.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

#### **`subcluster0.cpu6.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

#### **`subcluster0.cpu6.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: 0x123456

#### **`subcluster0.cpu6.semihosting-T32_HLT`**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

#### **`subcluster0.cpu6.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

#### **`subcluster0.cpu6.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

#### **`subcluster0.cpu6.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

### **`subcluster0.cpu6.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster0.cpu6.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu6.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu6.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster0.cpu6.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu6.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu6.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster0.cpu6.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster0.dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.dcache-maintenance\_latency**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.dcache-miss\_latency**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.dcache-prefetch\_enabled**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

Type: `bool`

Default value: `false`

**subcluster0.dcache-read\_access\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.dcache-size**

L1 D-Cache size in bytes.

Type: uint32\_t

Default value: 0x8000

**subcluster0.dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.dcache-write\_access\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.force\_zero\_PSTATE\_PAN**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: bool

Default value: false

**subcluster0.has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: bool

Default value: false

**subcluster0.has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: bool

Default value: true

**subcluster0.has\_dot\_product**

Possible values of this parameter are:- 1, feature not implemented.- 2, feature is implemented.

Type: uint8\_t

Default value: 2

**subcluster0.icache-hit\_latency**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

**subcluster0.icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.icache-size**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: `0x8000`

#### **`subcluster0.invalidate_code_cache_on_icache_cmo`**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: `uint8_t`

Default value: `0`

#### **`subcluster0.ptw_latency`**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.tlb_latency`**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.tlbi_stall_enabled`**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: `bool`

Default value: `false`

#### **`subcluster0.walk_cache_latency`**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.NUM_CORES`**

Number of cores per cluster.

Type: `uint8_t`

Default value: `1`

**subcluster1.cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**subcluster1.cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**subcluster1.cpu0.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu0.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster1.cpu0.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu0.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu0.VINITHI**

Reset value of SCTLR.V.

Type: `bool`



Default value: `false`

### **`subcluster1.cpu0.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu0.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu0.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu0.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu0.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu0.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu0.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

### **`subcluster1.cpu0.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu0.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu0.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu0.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu0.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

### **`subcluster1.cpu0.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

### **`subcluster1.cpu0.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu0.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu0.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

### **`subcluster1.cpu0.semihosting-T32_HLT`**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

### **`subcluster1.cpu0.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

### **`subcluster1.cpu0.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**`subcluster1.cpu0.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**`subcluster1.cpu0.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**`subcluster1.cpu0.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**`subcluster1.cpu0.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**`subcluster1.cpu0.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**`subcluster1.cpu0.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu0.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu0.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster1.cpu1.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: false

**subcluster1.cpu1.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: bool

Default value: false

**subcluster1.cpu1.CRYPTODISABLE**

Disable cryptographic features.

Type: bool

Default value: false

**subcluster1.cpu1.RVBARADDR**

Value of RVBAR\_ELx register.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu1.VINITHI**

Reset value of SCTLR.V.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu1.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu1.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to

the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu1.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

#### **`subcluster1.cpu1.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu1.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu1.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu1.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu1.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu1.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu1.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu1.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu1.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu1.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster1.cpu1.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t



Default value: 171

**subcluster1.cpu1.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu1.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu1.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu1.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu1.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu1.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu1.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: 0xf000000

**subcluster1.cpu1.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu1.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster1.cpu2.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: false

**subcluster1.cpu2.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: bool

Default value: false

**subcluster1.cpu2.CRYPTODISABLE**

Disable cryptographic features.

Type: bool

Default value: false

**subcluster1.cpu2.RVBARADDR**

Value of RVBAR\_ELx register.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu2.VINITHI**

Reset value of SCTL.R.V.

Type: `bool`

Default value: `false`

**subcluster1.cpu2.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu2.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

**subcluster1.cpu2.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster1.cpu2.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu2.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu2.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu2.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu2.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu2.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster1.cpu2.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

### **`subcluster1.cpu2.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **`subcluster1.cpu2.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **`subcluster1.cpu2.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster1.cpu2.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu2.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster1.cpu2.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster1.cpu2.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster1.cpu2.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu2.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu3.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster1.cpu3.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu3.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu3.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.VINITHI**

Reset value of SCTL.R.V.

Type: `bool`

Default value: `false`

**subcluster1.cpu3.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu3.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`



**subcluster1.cpu3.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster1.cpu3.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu3.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu3.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu3.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu3.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu3.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster1.cpu3.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

### **`subcluster1.cpu3.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **`subcluster1.cpu3.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **`subcluster1.cpu3.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster1.cpu3.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu3.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster1.cpu3.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster1.cpu3.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster1.cpu3.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu3.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster1.dcache-hit_latency`**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.dcache-maintenance_latency`**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.dcache-miss_latency`**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.dcache-prefetch_enabled`**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

Type: `bool`

Default value: `false`

#### **`subcluster1.dcache-read_access_latency`**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.dcache-read_latency`**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.dcache-size`**

L1 D-Cache size in bytes.

Type: `uint32_t`

Default value: `0x10000`

#### **`subcluster1.dcache-snoop_data_transfer_latency`**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.dcache-write_access_latency`**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.ext\_abort\_device\_read\_is\_sync**

Synchronous reporting of device-nGnRE read external aborts.

Type: bool

Default value: false

**subcluster1.ext\_abort\_device\_write\_is\_sync**

Synchronous reporting of device-nGnRE write external aborts.

Type: bool

Default value: false

**subcluster1.ext\_abort\_so\_read\_is\_sync**

Synchronous reporting of device-nGnRnE read external aborts.

Type: bool

Default value: false

**subcluster1.ext\_abort\_so\_write\_is\_sync**

Synchronous reporting of device-nGnRnE write external aborts.

Type: bool

Default value: false

**subcluster1.force\_zero\_PSTATE\_PAN**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: bool

Default value: false

**subcluster1.has\_coherent\_icache**

Whether icache invalidation to the point of unification is required for instruction to data coherence.  
true - Invalidate operations not required.

Type: bool

Default value: false

**subcluster1.has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: bool

Default value: false

**subcluster1.has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: bool

Default value: false

**subcluster1.has\_dot\_product**

Possible values of this parameter are:- 1, feature not implemented.- 2, feature is implemented.

Type: uint8\_t

Default value: 2

**subcluster1.icache-hit\_latency**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

**subcluster1.icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.icache-size**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: `0x10000`

**subcluster1.invalidate\_code\_cache\_on\_icache\_cmo**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.



Type: `uint8_t`

Default value: 0

### **`subcluster1.ptw_latency`**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

### **`subcluster1.tlb_latency`**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

### **`subcluster1.tlbi_stall_enabled`**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: `bool`

Default value: `false`

### **`subcluster1.treat-dcache-cmos-to-pou-as-nop`**

Whether dcache invalidation to the point of unification is required for instruction to data coherence. 0 - Invalidate ops required, 1 - Invalidate ops not required and cannot generate faults, 2 - Invalidate ops not required but can generate faults.

Type: `uint8_t`

Default value: 0

### **`subcluster1.walk_cache_latency`**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

## 3.60 ARMCortexA57CT

Defined in `LISA/ARMCortexA57CT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r1p1	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following parameters were added:

- `has_delayed_dbgreg`
- `has_delayed_sysreg`

### About ARMCortexA57CT

The per-core parameters are preceded by `cpun.`, where `n` identifies the core (0-3).

The cache latency parameters are only effective when you enable cache-state modeling. Timing annotation for transactions downstream of the cache and TLB models propagates through the models. This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (`crypto.dll` or `crypto.so`) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

The system designer decides whether a debug APB ties the external debug view with other system views. In the model, use a `PVBusDecoder` to direct traffic to the correct port, `dev_debug_s` or `memorymapped_debug_s`.

### Differences between the model and the RTL

This component has the following differences from the corresponding revision of the RTL implementation:

- The value of the AArch64 `PMCEID0_ELO` register, and the AArch32 alias of this register, differs in the model from the TRM value. The model value reflects the model counters.
- The mechanisms for setting the affinity fields of the MPIDR. The RTL has two ports:
  - `CLUSTERIDAFF1[7:0]`.
  - `CLUSTERIDAFF2[7:0]`. `AFF1` sets the value of MPIDR bits[15:8] and `AFF2` sets the value of MPIDR bits[23:16]. In contrast, the model has a single `CLUSTER_ID` port. This difference allows the setting of bits[23:8] of the MPIDR using bits[15:0] of the `CLUSTER_ID` value.
- The memory mapped debug registers have a view for cores and a view for external debug agents. In the model, these views require two `PVBus` ports. In hardware, the system designer decides how the implementation differentiates the views.
- In the model, a single peer event port combines the functionality of the `eventi` and `evento` signals in the RTL.

- The Generic Timers are Programmer's View (PV) level abstractions: a model-specific protocol connects the cntvalueb port to the MemoryMappedCounterModule.
- The GIC CPU Interface is a PV level abstraction: a model-specific protocol connects the GIC CPU Interface to the GIC Distributor.
- The CoreSight Cross Trigger Interface (CTI) is a PV level abstraction: the interface is a model-specific one.
- The model has no mechanism to read the internal memory that the Cache and TLB structures use, through the implementation defined region of the system coprocessor interface. This memory includes the RAM Index Register, IL1DATA Registers, DL1DATA Registers, and associated functionality.
- The model does not implement:
  - ETM registers.
  - The PMUEVENT bus.
  - The WARMRESETREQ signal. However, the warm reset code sequence (see the section Code sequence to request a Warm reset as a result of RMR\_ELx.RR in the Arm Architecture Reference Manual for A-profile architecture) makes the model simulate a warm reset of the core.
  - The PMUSNAPSHOTREQ and PMUSNAPSHOTACK signals.
  - The EXTERRIRQ and INTERRIRQ signals.
  - Processor dynamic-retention signals.
  - The SYSBARDISABLE signal.
  - This model has a variable number of cores per cluster, specified using the `NUM_CORES` parameter.
  - The DBGPWRDUP, DBGPWRUPREQ, DBGNOPWRDWN, and DBGRSTREQ debug power management signals.
  - ECC and parity schemes are hardware-specific so are not supported.

## Iris and MTI instances for ARM CortexA57CT

This model has the following Iris instances:

Name	Instance type
ARM CortexA57CT	Cluster_ARM_Cortex-A57
ARM CortexA57CT.AMU	PVBusLogger
ARM CortexA57CT.AMU.mapper	PVBusMapper
ARM CortexA57CT.DAP	PVBusLogger
ARM CortexA57CT.DAP.mapper	PVBusMapper
ARM CortexA57CT.DSU	DSU
ARM CortexA57CT.DSU.mpam_busslave	PVBusSlave
ARM CortexA57CT.MMAP	PVBusLogger
ARM CortexA57CT.MMAP.mapper	PVBusMapper
ARM CortexA57CT.RAS	PVBusLogger

Name	Instance type
ARMCortexA57CT.RAS.mapper	PVBusMapper
ARMCortexA57CT.acp_mapper	PVBusMapper
ARMCortexA57CT.cpu0	ARM_Cortex-A57
ARMCortexA57CT.cpu0.UTLB	TLB
ARMCortexA57CT.cpu0.dtlb	TLB
ARMCortexA57CT.cpu0.l1dcache	PVCache
ARMCortexA57CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA57CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA57CT.cpu0.l1icache	PVCache
ARMCortexA57CT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA57CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA57CT.ext_bus	PVBusLogger
ARMCortexA57CT.ext_bus.mapper	PVBusMapper
ARMCortexA57CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA57CT.global_debug_rom	debug_rom
ARMCortexA57CT.l2_cache	PVCache
ARMCortexA57CT.l2_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA57CT.l2_cache.upstream[Z] (where Z = 0-16)	PVBusSlave
ARMCortexA57CT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

Name	Component type
ARMCortexA57CT.AMU	PVBusLogger
ARMCortexA57CT.AMU.mapper	PVBusMapper
ARMCortexA57CT.DAP	PVBusLogger
ARMCortexA57CT.DAP.mapper	PVBusMapper
ARMCortexA57CT.DSU	DSU
ARMCortexA57CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA57CT.MMAP	PVBusLogger
ARMCortexA57CT.MMAP.mapper	PVBusMapper
ARMCortexA57CT.RAS	PVBusLogger
ARMCortexA57CT.RAS.mapper	PVBusMapper
ARMCortexA57CT.acp_mapper	PVBusMapper
ARMCortexA57CT.cpu0	ARM_Cortex-A57
ARMCortexA57CT.cpu0.UTLB	TLB
ARMCortexA57CT.cpu0.l1dcache	PVCache
ARMCortexA57CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA57CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA57CT.cpu0.l1icache	PVCache
ARMCortexA57CT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster

Name	Component type
ARMCortexA57CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA57CT.ext_bus	PVBusLogger
ARMCortexA57CT.ext_bus.mapper	PVBusMapper
ARMCortexA57CT.gic_cpuif_decoder_cluster	GCv3CPUInterfaceDecoder
ARMCortexA57CT.l2_cache	PVCache
ARMCortexA57CT.l2_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA57CT.l2_cache.upstream[Z] (where Z = 0-16)	PVBusSlave
ARMCortexA57CT.l2_flusher	AsyncCacheFlushUnit

## Ports for ARMCortexA57CT

Port	Direction	Protocol	Description
aa64naa32	slave	Signal	Register width after reset.
acp_s	slave	PVBus	AXI ACP slave port.
broadcastcachemaint	slave	Signal	Enable broadcasting of cache maintenance operations to downstream caches.
broadcastinner	slave	Signal	Enable broadcasting of Inner Shareable transactions.
broadcastouter	slave	Signal	Enable broadcasting of Outer Shareable transactions.
cfgend	slave	Signal	This signal is for EE bit initialisation.
cfgsdisable	slave	Signal	This signal disables write access to some secure Interrupt Controller registers.
cfgte	slave	Signal	This signal provides default exception handling state.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
clrexmonack	master	Signal	Acknowledge handshake signal for the clrexmonreq signal
clrexmonreq	slave	Signal	Signals the clearing of an external global exclusive monitor
clusterid	slave	Value	The port sets the value of the affinity levels 1 and 2; bits [23:16] and [15:8] of the MPIDR.
CNTHPIRQ	master	Signal	The per-EL counter signal.
CNTPNSIRQ	master	Signal	The per-EL counter signal.
CNTPSIRQ	master	Signal	The per-EL counter signal.
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.
CNTVIRQ	master	Signal	The per-EL counter signal.
commirq	master	Signal	Interrupt signal from debug communications channel.
commrxd	master	Signal	Receive portion of Data Transfer Register full.
commtxd	master	Signal	Transmit portion of Data Transfer Register empty.
cp15sdisable	slave	Signal	This signal disables write access to some system control processor registers.
cpuporeset	slave	Signal	Power on reset. Initializes all the processor logic, including debug logic.
cryptodisable	slave	Signal	Disable cryptography extensions after reset.

Port	Direction	Protocol	Description
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
ctidbgirq	master	Signal	Cross Trigger Interface (CTI) interrupt trigger output.
dbgack	master	Signal	External debug interface.
dbgen	slave	Signal	External debug interface.
dbgnopwrdown	master	Signal	These signals relate to core power down.
dbgpwrupreq	master	Signal	These signals relate to core power down.
dev_debug_s	slave	PVBus	External debug interface.
edbgrq	slave	Signal	External debug interface.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
l2flushdone	master	Signal	Flush of L2 memory system complete
l2flushreq	slave	Signal	Request flush of L2 memory system.
l2reset	slave	Signal	Reset the shared L2 memory system controller.
memorymapped_debug_s	slave	PVBus	External debug interface.
niden	slave	Signal	External debug interface.
periphbase	slave	Value_64	This port sets the base address of private peripheral region.
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	master	PVBus	The core will generate bus requests on this port.
rei	slave	Signal	Individual processor RAM Error Interrupt signal input.
reset	slave	Signal	Raising this signal will put the core into reset mode.
romaddr	slave	Value_64	Debug ROM base address.
romaddrv	slave	Signal	Debug ROM base address valid.
rvbaraddr	slave	Value_64	Reset vector base address.
sei	slave	Signal	Per core System Error physical pins.
smpen	master	Signal	This signals AMP or SMP mode for each core.
spiden	slave	Signal	External debug interface.
spniden	slave	Signal	External debug interface.
standbywfe	master	Signal	This signal indicates if a core is in WFE state.
standbywfi	master	Signal	This signal indicates if a core is in WFI state.
standbywfil2	master	Signal	Indicate that all the individual processors and the L2 systems are in a WFI state.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.

Port	Direction	Protocol	Description
vfiq	slave	Signal	Virtual FIQ.
vinithi	slave	Signal	This signal controls of the location of the exception vectors at reset.
virq	slave	Signal	Virtual IRQ.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Processor Virtual System Error Interrupt request.

## Parameters for ARMCortexA57CT

### **cpuX.AA64nAA32**

Type: `bool`

Default value: `true`

### **cpuX.CFGEND**

Type: `bool`

Default value: `false`

### **cpuX.CFGTE**

Type: `bool`

Default value: `false`

### **cpuX.CP15SDISABLE**

Type: `bool`

Default value: `false`

### **cpuX.CRYPTODISABLE**

Type: `bool`

Default value: `false`

### **cpuX.RVBARADDR**

Type: `uint64_t`

Default value: `0`

### **cpuX.VINITHI**

Type: `bool`

Default value: `false`

**cpuX.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter trace\_special\_hlt\_imm16.

Type: `bool`

Default value: `false`

**cpuX.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint64_t`

Default value: `256`

**cpuX.min\_sync\_level**

force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint32_t`

Default value: `0`

**cpuX.semihosting-A32\_HLT**

Type: `uint32_t`

Default value: `0xF000`

**cpuX.semihosting-A64\_HLT**

Type: `uint32_t`

Default value: `0xF000`

**cpuX.semihosting-ARM\_SVC**

Type: `uint32_t`

Default value: `0x123456`

**cpuX.semihosting-T32\_HLT**

Type: `uint32_t`

Default value: `0x3c`

**cpuX.semihosting-Thumb\_SVC**

Type: `uint32_t`

Default value: `0xAB`



**cpuX.semihosting-cmd\_line**

Type: string

Default value: ""

**cpuX.semihosting-cwd**

Type: string

Default value: ""

**cpuX.semihosting-enable**

Type: bool

Default value: true

**cpuX.semihosting-heap\_base**

Type: uint32\_t

Default value: 0x0

**cpuX.semihosting-heap\_limit**

Type: uint32\_t

Default value: 0x0F000000

**cpuX.semihosting-stack\_base**

Type: uint32\_t

Default value: 0x10000000

**cpuX.semihosting-stack\_limit**

Type: uint32\_t

Default value: 0x0F000000

**cpuX.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**cpuX.vfp-enable\_at\_reset**

Type: bool

Default value: false

**BROADCASTCACHEMAINT**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

Type: `bool`

Default value: `true`

**BROADCASTINNER**

Enable broadcasting of Inner Shareable transactions. The broadcastinner signal will override this value if used.

Type: `bool`

Default value: `true`

**BROADCASTOUTER**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

Type: `bool`

Default value: `true`

**CLUSTER\_ID**

Type: `uint32_t`

Default value: `0`

**DBGROMADDR**

Type: `uint64_t`

Default value: `0x22000000`

**DBGROMADDRV**

Type: `bool`

Default value: `true`

**GICDISABLE**

Type: `bool`

Default value: `true`

**NUM\_CORES**

Number of cores in cluster.

Type: `uint8_t`

Default value: 1

### **PERIPHBASE**

Type: `uint64_t`

Default value: 0x13080000

### **bus\_type**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

Type: `uint32_t`

Default value: 0

### **cpi\_div**

divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: 1

### **cpi\_mul**

multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: 1

### **dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit.

Type: `uint64_t`

Default value: 0

### **dcache-maintenance\_latency**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

### **dcache-miss\_latency**

L1 D-Cache timing annotation latency for miss.

Type: `uint64_t`

Default value: 0

### **`dcache-read_access_latency`**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (based on the size of `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

### **`dcache-read_latency`**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

### **`dcache-snoop_data_transfer_latency`**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

### **`dcache-state_modelled`**

Type: `bool`

Default value: `false`

### **`dcache-write_access_latency`**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (based on the size of `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

### **`dcache-write_latency`**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

### **has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

### **has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

### **icache-hit\_latency**

L1 I-Cache timing annotation latency for hit.

Type: `uint64_t`

Default value: 0

### **icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

### **icache-miss\_latency**

L1 I-Cache timing annotation latency for miss.

Type: `uint64_t`

Default value: 0

### **icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

**icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (based on the size of icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0

**icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. This is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0

**icache-state\_modelled**

Type: bool

Default value: false

**l2cache-hit\_latency**

L2 Cache timing annotation latency for hit.

Type: uint64\_t

Default value: 0

**l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0

**l2cache-miss\_latency**

L2 Cache timing annotation latency for miss.

Type: uint64\_t

Default value: 0

**l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access (based on the size of l2cache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0

**l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0

**l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0

**l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0

**l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access (based on the size of l2cache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0

### **l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0

### **ptw\_latency**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint32\_t

Default value: 0

### **tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint32\_t

Default value: 0

## 3.61 ARMCortexA65AECT

Defined in LISA/ARMCortexA65AECT.lisa.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following parameters were added:

- has\_delayed\_dbgreg
- has\_delayed\_sysreg

### About ARMCortexA65AECT

The model supports the following features:

- DynamIQ (DSU) system registers.
- Per-core L2 cache.



- A P-Channel for the cluster and for each core.
- Optional peripheral port.
- L3 cache partition.
- Per-core clock.
- `BROADCASTPERSIST` pin.

The model does not support the following features:

- `BROADCASTCACHEMAINTPOU` pin.
- `COREINSTRRET`, `COREINSTRRUN`, or `nPMBIRQ` signals.
- DynamIQ features that are irrelevant to the programmers' view simulation, for example:
  - Automatic CPU retention mode.
  - Level-3 Cache RAM retention.
- 256-bit wide output transactions.
- ECC and parity schemes are hardware-specific so are not supported.
- Self-test features (MBIST).
- Latency configuration.
- Snoop filtering.
- Debug power management signals `DBGPWRDUP` and `DBGIRSTREQ` are not implemented, but `DBGPWRUPREQ` and `DBGNOPWRDWN` are implemented.
- Cache stashing capability.
- Each thread has its own L1Cache and L2Cache instead of a common cache that is shared by all threads of the core.
- Per-thread parameters are not yet supported but signals are implemented.
- Split/Lock is supported but with the limitations described in the AE-specific features implemented section.

This model has a variable number of cores per cluster, specified using the `NUM_CORES` parameter.

The per-core parameters are preceded by `cpun.`, where `n` identifies the core (0-7).

This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (`crypto.dll` or `crypto.so`) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

### AE-specific features implemented

The only AE-specific features implemented in the model are Split-Lock and Lock-Step, with the following limitations:

- The implementation of Split-Lock and Lock-Step in the model is simply a halving of the number of cores available.

- Lock-step is implemented by a parameter that can be true or false for the entire execution only, rather than by a signal that can be enabled during execution.

As the model does not implement true Split-Lock functionality, the ports and parameters are contiguous. For example, for a 4-core CPU with Split-Lock enabled, `cpu0` and `cpu1` identify the available cores and associated ports, not `cpu0` and `cpu2`.

- Hybrid mode is not modeled in the DSU.

## Iris and MTI instances for ARMCortexA65AECT

This model has the following Iris instances:

Name	Instance type
ARMCortexA65AECT	Cluster_ARM_Cortex-A65AE
ARMCortexA65AECT.AMU	PVBusLogger
ARMCortexA65AECT.AMU.mapper	PVBusMapper
ARMCortexA65AECT.DAP	PVBusLogger
ARMCortexA65AECT.DAP.mapper	PVBusMapper
ARMCortexA65AECT.DSU	DSU
ARMCortexA65AECT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA65AECT.DSU.mpam_busslave	PVBusSlave
ARMCortexA65AECT.DSU.shared_cache	PVCache
ARMCortexA65AECT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA65AECT.DSU.shared_cache.upstream[Y] (where Y = 0-5)	PVBusSlave
ARMCortexA65AECT.MMAP	PVBusLogger
ARMCortexA65AECT.MMAP.mapper	PVBusMapper
ARMCortexA65AECT.RAS	PVBusLogger
ARMCortexA65AECT.RAS.mapper	PVBusMapper
ARMCortexA65AECT.cpuY.dtlb (where Y = 0-1)	TLB
ARMCortexA65AECT.cpuY.threadZ (where Y = 0-1; Z = 0-1)	ARM_Cortex-A65AE
ARMCortexA65AECT.cpuY.threadZ.UTLB (where Y = 0-1; Z = 0-1)	TLB
ARMCortexA65AECT.cpuY.thread0.l1dcache (where Y = 0-1)	PVCache
ARMCortexA65AECT.cpuY.thread0.l1dcache.downstream[0].pvbusmaster (where Y = 0-1)	PVBusMaster
ARMCortexA65AECT.cpuY.thread0.l1dcache.upstream[0] (where Y = 0-1)	PVBusSlave
ARMCortexA65AECT.cpuY.thread0.l1icache (where Y = 0-1)	PVCache
ARMCortexA65AECT.cpuY.thread0.l1icache.downstream[0].pvbusmaster (where Y = 0-1)	PVBusMaster
ARMCortexA65AECT.cpuY.thread0.l1icache.upstream[0] (where Y = 0-1)	PVBusSlave
ARMCortexA65AECT.cpuY.thread0.l2cache (where Y = 0-1)	PVCache
ARMCortexA65AECT.cpuY.thread0.l2cache.downstream[0].pvbusmaster (where Y = 0-1)	PVBusMaster
ARMCortexA65AECT.cpuY.thread0.l2cache.upstream[V] (where Y = 0-1; V = 0-1)	PVBusSlave
ARMCortexA65AECT.ext_bus	PVBusLogger
ARMCortexA65AECT.ext_bus.mapper	PVBusMapper
ARMCortexA65AECT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA65AECT.global_debug_rom	debug_rom

This model has the following MTI trace components:

Name	Component type
ARMCortexA65AECT.AMU	PVBusLogger
ARMCortexA65AECT.AMU.mapper	PVBusMapper
ARMCortexA65AECT.DAP	PVBusLogger
ARMCortexA65AECT.DAP.mapper	PVBusMapper
ARMCortexA65AECT.DSU	DSU
ARMCortexA65AECT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA65AECT.DSU.mpam_busslave	PVBusSlave
ARMCortexA65AECT.DSU.shared_cache	PVCache
ARMCortexA65AECT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA65AECT.DSU.shared_cache.upstream[Y] (where Y = 0–5)	PVBusSlave
ARMCortexA65AECT.MMAP	PVBusLogger
ARMCortexA65AECT.MMAP.mapper	PVBusMapper
ARMCortexA65AECT.RAS	PVBusLogger
ARMCortexA65AECT.RAS.mapper	PVBusMapper
ARMCortexA65AECT.cpuY.threadZ (where Y = 0–1; Z = 0–1)	ARM_Cortex-A65AE
ARMCortexA65AECT.cpuY.threadZ.UTLB (where Y = 0–1; Z = 0–1)	TLB
ARMCortexA65AECT.cpuY.thread0.l1dcache (where Y = 0–1)	PVCache
ARMCortexA65AECT.cpuY.thread0.l1dcache.downstream[0].pvbusmaster (where Y = 0–1)	PVBusMaster
ARMCortexA65AECT.cpuY.thread0.l1dcache.upstream[0] (where Y = 0–1)	PVBusSlave
ARMCortexA65AECT.cpuY.thread0.l1icache (where Y = 0–1)	PVCache
ARMCortexA65AECT.cpuY.thread0.l1icache.downstream[0].pvbusmaster (where Y = 0–1)	PVBusMaster
ARMCortexA65AECT.cpuY.thread0.l1icache.upstream[0] (where Y = 0–1)	PVBusSlave
ARMCortexA65AECT.cpuY.thread0.l2cache (where Y = 0–1)	PVCache
ARMCortexA65AECT.cpuY.thread0.l2cache.downstream[0].pvbusmaster (where Y = 0–1)	PVBusMaster
ARMCortexA65AECT.cpuY.thread0.l2cache.upstream[V] (where Y = 0–1; V = 0–1)	PVBusSlave
ARMCortexA65AECT.ext_bus	PVBusLogger
ARMCortexA65AECT.ext_bus.mapper	PVBusMapper
ARMCortexA65AECT.gic_cpuif_decoder_cluster	GIcV3CPUInterfaceDecoder

## Ports for ARMCortexA65AECT

Port	Direction	Protocol	Description
acp_s	slave	PVBus	AXI ACP slave port
AENDMP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
ASTARTMP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	slave	Signal	CHI defined pins.
broadcastcachemaint	slave	Signal	ACE defined pins.
broadcastouter	slave	Signal	ACE defined pins.

Port	Direction	Protocol	Description
broadcastpersist	slave	Signal	CHI defined pins.
cfgend	slave	Signal	This signal is for EE bit initialisation.
cfgte	slave	Signal	This signal provides default exception handling state.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
clusterid	slave	Value	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set ManagerID64 of the core.
clusterpmuirq	master	Signal	DynamiQ pmu irq
CNTHPIRQ	master	Signal	Timer signals to SOC.
CNTHVIRQ	master	Signal	Timer signals to SOC.
CNTPNSIRQ	master	Signal	Timer signals to SOC.
CNTPSIRQ	master	Signal	Timer signals to SOC.
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.
CNTVIRQ	master	Signal	Timer signals to SOC.
commirq	master	Signal	Interrupt signal from debug communications channel.
core_clk_in	slave	ClockSignal	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
cpuporeset	slave	Signal	Power on reset. Initializes all the processor logic, including debug logic.
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
ctidbgirq	master	Signal	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	slave	Signal	External debug interface.
dbgnopwrdown	master	Signal	These signals relate to core power down.
dbgprupreq	master	Signal	Debug power up request.
dev_debug_s	slave	PVBus	External debug interface.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port per thread.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	slave	PVBus	External debug interface.
niden	slave	Signal	External debug interface.
pchannel_cluster	slave	PChannel	PChannel for cluster.

Port	Direction	Protocol	Description
pchannel_core	slave	PChannel	PChannels for cores
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	master	PVBus	The core will generate bus requests on this port.
pvbus_periph_m	master	PVBus	The core can generate peripheral bus request on this port.
rei	slave	Signal	Per core RAM Error Interrupt.
reset	slave	Signal	Raising this signal will put the core into reset mode.
rvbaraddr	slave	Value_64	Reset vector base address.
sci_m	master	SystemCoherencyInterface	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei	slave	Signal	Per core virtual System Error physical pins.
spiden	slave	Signal	External debug interface.
spniden	slave	Signal	External debug interface.
sporeset	slave	Signal	A single cluster-wide power on reset signal for all resettable registers in DynamIQ.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfiq	slave	Signal	Virtualised FIQ.
vinithi	slave	Signal	This signal controls of the location of the exception vectors at reset.
virq	slave	Signal	Virtualised IRQ.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Per core virtual System Error physical pins.

## Parameters for ARM Cortex A65AECT

### **cpuX.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **cpuX.enable\_single\_thread\_at\_reset**

Enable single thread after reset and keep other thread in reset.

Type: `bool`

Default value: `false`

**cpuX.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter trace\_special\_hlt\_imm16.

Type: bool

Default value: false

**cpuX.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **cpuX.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x40000

### **cpuX.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **cpuX.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **cpuX.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **cpuX.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **cpuX.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**cpuX.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**cpuX.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**cpuX.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**cpuX.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**cpuX.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**cpuX.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**cpuX.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string



Default value: N/A

**cpuX.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**cpuX.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**cpuX.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**cpuX.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**cpuX.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**cpuX.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**cpuX.thread0.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian.

Type: `bool`

Default value: `false`

**cpuX.thread0.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**cpuX.thread0.MPIDR-override**

Override MPIDR value. A nonzero value will override the MT,cluster,cpu,thread ID bits in MPIDR.

Type: `uint64_t`

Default value: `0x0`

**cpuX.thread0.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**cpuX.thread0.VINITHI**

Reset value of SCTLR.V.

Type: `bool`

Default value: `false`

**cpuX.thread1.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian.

Type: `bool`

Default value: `false`

**cpuX.thread1.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**cpuX.thread1.MPIDR-override**

Override MPIDR value. A nonzero value will override the MT,cluster,cpu,thread ID bits in MPIDR.

Type: `uint64_t`

Default value: `0x0`

**cpuX.thread1.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**cpuX.thread1.VINITHI**

Reset value of SCTLR.V.

Type: `bool`

Default value: `false`

**cpuX.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**cpuX.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**cpuX.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**BROADCASTATOMIC**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`

Default value: `true`

**BROADCASTCACHEMAINT**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

Type: `bool`

Default value: `true`

**BROADCASTOUTER**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

Type: `bool`

Default value: `true`

**BROADCASTPERSIST**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: `bool`

Default value: `true`

**CLUSTER\_ID**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the ManagerID64 of the core.

Type: `uint32_t`

Default value: `0x0`

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain even when cache state modelling is disabled. Possible values are:- 0 = All CMOs are broadcast if architecturally required- 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required.

Type: `uint8_t`

Default value: `1`

**GICDISABLE**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`

Default value: `true`

**NUM\_CORES**

Number of cores per cluster.

Type: `uint8_t`

Default value: 2

**cluster\_patch\_level**

Cosmetic change to patch number in CLUSTERIDR. Corresponds to the Y in rXpY.

Type: `uint8_t`

Default value: 0

**cluster\_revision\_number**

Cosmetic change to revision number in CLUSTERIDR, Corresponds to the X in rXpY.

Type: `uint8_t`

Default value: 0

**cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: 0x1

**cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: 0x1

**dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`dcache-maintenance_latency`**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`dcache-miss_latency`**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`dcache-prefetch_enabled`**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

Type: `bool`

Default value: `false`

### **`dcache-read_access_latency`**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`dcache-read_latency`**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-size**

L1 D-Cache size in bytes.

Type: `uint32_t`

Default value: `0x10000`

**dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-state\_modelled**

Set whether D-cache has stateful implementation.

Type: `bool`

Default value: `false`

**dcache-write\_access\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**default\_opmode**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

Type: `uint8_t`

Default value: 4

**diagnostics**

Enable DynaMIQ diagnostic messages.

Type: `bool`

Default value: `false`

**enable\_lock\_step**

Whether the core is configured in Dual Core Lock Step mode (FEAT\_DCLS).

Type: `bool`

Default value: `false`

**enable\_simulation\_performance\_optimizations**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of `stage12_tlb_size` parameter to 1024).

Type: `bool`

Default value: `true`

**force\_zero\_PSTATE\_PAN**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

**has\_acp**

If true, Accelerator Coherency Port is configured.

Type: `bool`

Default value: `false`

**has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`



**has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

**has\_dot\_product**

Possible values of this parameter are:- 1, feature not implemented.- 2, feature is implemented.

Type: `uint8_t`

Default value: 2

**has\_peripheral\_port**

If true, additional AXI peripheral port is configured.

Type: `bool`

Default value: `false`

**icache-hit\_latency**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

Type: `bool`

Default value: `false`

**icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when icache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**icache-size**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: `0x10000`

**icache-state\_modelled**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

**invalidate\_code\_cache\_on\_icache\_cmo**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: `uint8_t`

Default value: 0

### **l3cache-hit\_latency**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **l3cache-maintenance\_latency**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **l3cache-miss\_latency**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **l3cache-read\_access\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size l3cache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **l3cache-read\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. l3cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**l3cache-size**

L3 Cache size in bytes.

Type: `uint32_t`

Default value: `0x400000`

**l3cache-snoop\_data\_transfer\_latency**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-snoop\_issue\_latency**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-write\_access\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size `l3cache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-write\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l3cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**pchannel\_treat\_simreset\_as\_poreset**

Register core as ON state to cluster with simulation reset.

Type: `bool`

Default value: `false`

**periph\_address\_end**

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

Type: uint64\_t

Default value: 0x0

**periph\_address\_start**

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

Type: uint64\_t

Default value: 0x0

**ptw\_latency**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

**reported\_patch\_level**

Purely cosmetic patch level value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: int8\_t

Default value: -1

**reported\_revision\_number**

Purely cosmetic revision number value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: int8\_t

Default value: -1

**tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

**tlbi\_stall\_enabled**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: `bool`

Default value: `false`

**walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

3.62 **ARMCortexA65AECT\_CortexA76AECT**

Defined in `LISA/ARMCortexA65AECT_CortexA76AECT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
CortexA65AE r0p0	Full support
CortexA76AE r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

**Changes in 11.31.15**

The following parameters were added:

- `subcluster0.has_delayed_dbgreg`
- `subcluster0.has_delayed_sysreg`
- `subcluster1.has_delayed_dbgreg`
- `subcluster1.has_delayed_sysreg`

**About ARMCortexA65AECT\_CortexA76AECT**

The number of cores in each subcluster is configurable using the following parameters:

**subcluster0.NUM\_CORES**

Possible values are 2-6 (ARMCortexA65AECT).

**subcluster1.NUM\_CORES**

Possible values are 2-4 (ARMCortexA76AECT).

The total number of cores in the cluster cannot exceed 8.

Port arrays are expanded to the width needed for the maximum number of cores in each subcluster. Use the following port array indexes:

- `port_name[0-11]` for cores in subcluster0.

- port\_name[0] is a port for subcluster0.cpu0.thread0
- port\_name[1] is a port for subcluster0.cpu0.thread1
- port\_name[2] is a port for subcluster0.cpu1.thread0
- port\_name[12-15] for cores in subcluster1.



All instances in the Master cross trigger matrix port array, `cti[16]` must be connected, regardless of the `NUM_CORES` value used.

Core-specific parameters have the following prefixes:

- subcluster0.cpu0 to subcluster0.cpu5 identify cores in subcluster0.
- subcluster1.cpu0 to subcluster1.cpu3 identify cores in subcluster1.

For information about the cores in this model, see:

- [ARMCortexA65AECT](#)
- [ARMCortexA76AECT](#)

## Iris and MTI instances for ARMCortexA65AECT\_CortexA76AECT

This model has the following Iris instances:

Name	Instance type
ARMCortexA65AECT_CortexA76AECT	Cluster_ARM_Cortex-A65AE_Cortex-A76AE
ARMCortexA65AECT_CortexA76AECT.AMU	PVBusLogger
ARMCortexA65AECT_CortexA76AECT.AMU.mapper	PVBusMapper
ARMCortexA65AECT_CortexA76AECT.DAP	PVBusLogger
ARMCortexA65AECT_CortexA76AECT.DAP.mapper	PVBusMapper
ARMCortexA65AECT_CortexA76AECT.DSU	DSU
ARMCortexA65AECT_CortexA76AECT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA65AECT_CortexA76AECT.DSU.mpam_busslave	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.DSU.shared_cache	PVCache
ARMCortexA65AECT_CortexA76AECT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA65AECT_CortexA76AECT.DSU.shared_cache.upstream[Z] (where Z = 0-9)	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.MMAP	PVBusLogger
ARMCortexA65AECT_CortexA76AECT.MMAP.mapper	PVBusMapper
ARMCortexA65AECT_CortexA76AECT.RAS	PVBusLogger
ARMCortexA65AECT_CortexA76AECT.RAS.mapper	PVBusMapper
ARMCortexA65AECT_CortexA76AECT.ext_bus	PVBusLogger
ARMCortexA65AECT_CortexA76AECT.ext_bus.mapper	PVBusMapper
ARMCortexA65AECT_CortexA76AECT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

Name	Instance type
ARMCortexA65AECT_CortexA76AECT.global_debug_rom	debug_rom
ARMCortexA65AECT_CortexA76AECT.subcluster0	Subcluster_ARM_Cortex-A65AE
ARMCortexA65AECT_CortexA76AECT.subclusterZ.cpuU.dtlb (where $Z = 0-1$ ; $U = 0-3$ )	TLB
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpuU.threadV (where $U = 0-1$ ; $V = 0-1$ )	ARM_Cortex-A65AE
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpuU.threadV.UTLB (where $U = 0-1$ ; $V = 0-1$ )	TLB
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpuU.thread0.l1dcache (where $U = 0-1$ )	PVCache
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpuU.thread0.l1dcache.downstream[0].pvbusmaster (where $U = 0-1$ )	PVBusMaster
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpuU.thread0.l1dcache.upstream[0] (where $U = 0-1$ )	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpuU.thread0.l1licache (where $U = 0-1$ )	PVCache
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpuU.thread0.l1licache.downstream[0].pvbusmaster (where $U = 0-1$ )	PVBusMaster
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpuU.thread0.l1licache.upstream[0] (where $U = 0-1$ )	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpuU.thread0.l2cache (where $U = 0-1$ )	PVCache
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpuU.thread0.l2cache.downstream[0].pvbusmaster (where $U = 0-1$ )	PVBusMaster
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpuU.thread0.l2cache.upstream[A] (where $U = 0-1$ ; $A = 0-1$ )	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.subcluster1	Subcluster_ARM_Cortex-A76AE
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpuU (where $U = 0-1$ )	ARM_Cortex-A76AE
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpuU.UTLB (where $U = 0-1$ )	TLB
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpuU.l1dcache (where $U = 0-1$ )	PVCache
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpuU.l1dcache.downstream[0].pvbusmaster (where $U = 0-1$ )	PVBusMaster
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpuU.l1dcache.upstream[0] (where $U = 0-1$ )	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpuU.l1licache (where $U = 0-1$ )	PVCache
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpuU.l1licache.downstream[0].pvbusmaster (where $U = 0-1$ )	PVBusMaster
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpuU.l1licache.upstream[0] (where $U = 0-1$ )	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpuU.l2cache (where $U = 0-1$ )	PVCache
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpuU.l2cache.downstream[0].pvbusmaster (where $U = 0-1$ )	PVBusMaster
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpuU.l2cache.upstream[W] (where $U = 0-1$ ; $W = 0-1$ )	PVBusSlave

This model has the following MTI trace components:



Name	Component type
ARMCortexA65AECT_CortexA76AECT.AMU	PVBusLogger
ARMCortexA65AECT_CortexA76AECT.AMU.mapper	PVBusMapper
ARMCortexA65AECT_CortexA76AECT.DAP	PVBusLogger
ARMCortexA65AECT_CortexA76AECT.DAP.mapper	PVBusMapper
ARMCortexA65AECT_CortexA76AECT.DSU	DSU
ARMCortexA65AECT_CortexA76AECT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA65AECT_CortexA76AECT.DSU.mpam_busslave	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.DSU.shared_cache	PVCache
ARMCortexA65AECT_CortexA76AECT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA65AECT_CortexA76AECT.DSU.shared_cache.upstream[Z] (where Z = 0-9)	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.MMAP	PVBusLogger
ARMCortexA65AECT_CortexA76AECT.MMAP.mapper	PVBusMapper
ARMCortexA65AECT_CortexA76AECT.RAS	PVBusLogger
ARMCortexA65AECT_CortexA76AECT.RAS.mapper	PVBusMapper
ARMCortexA65AECT_CortexA76AECT.ext_bus	PVBusLogger
ARMCortexA65AECT_CortexA76AECT.ext_bus.mapper	PVBusMapper
ARMCortexA65AECT_CortexA76AECT.gic_cpuif_decoder_cluster	GIcV3CPUInterfaceDecoder
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpuU.threadV (where U = 0-1; V = 0-1)	ARM_Cortex-A65AE
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpuU.threadV.UTLB (where U = 0-1; V = 0-1)	TLB
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpuU.thread0.l1dcache (where U = 0-1)	PVCache
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpuU.thread0.l1dcache.downstream[0].pvbusmaster (where U = 0-1)	PVBusMaster
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpuU.thread0.l1dcache.upstream[0] (where U = 0-1)	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpuU.thread0.l1licache (where U = 0-1)	PVCache
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpuU.thread0.l1licache.downstream[0].pvbusmaster (where U = 0-1)	PVBusMaster
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpuU.thread0.l1licache.upstream[0] (where U = 0-1)	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpuU.thread0.l2cache (where U = 0-1)	PVCache
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpuU.thread0.l2cache.downstream[0].pvbusmaster (where U = 0-1)	PVBusMaster
ARMCortexA65AECT_CortexA76AECT.subcluster0.cpuU.thread0.l2cache.upstream[A] (where U = 0-1; A = 0-1)	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpuU (where U = 0-1)	ARM_Cortex-A76AE
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpuU.UTLB (where U = 0-1)	TLB
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpuU.l1dcache (where U = 0-1)	PVCache
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpuU.l1dcache.downstream[0].pvbusmaster (where U = 0-1)	PVBusMaster
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpuU.l1dcache.upstream[0] (where U = 0-1)	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpuU.l1licache (where U = 0-1)	PVCache

Name	Component type
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpuU.l1icache.downstream[0].pvbusmaster (where $U = 0-1$ )	PVBusMaster
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpuU.l1icache.upstream[0] (where $U = 0-1$ )	PVBusSlave
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpuU.l2cache (where $U = 0-1$ )	PVCache
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpuU.l2cache.downstream[0].pvbusmaster (where $U = 0-1$ )	PVBusMaster
ARMCortexA65AECT_CortexA76AECT.subcluster1.cpuU.l2cache.upstream[W] (where $U = 0-1$ ; $W = 0-1$ )	PVBusSlave

### Ports for ARMCortexA65AECT\_CortexA76AECT

Port	Direction	Protocol	Description
acp_s	slave	PVBus	AXI ACP slave port
AENDMP	slave	Value_64	DynamiQ port to obtain end address of valid peripheral address range (exclusive).
ASTARTMP	slave	Value_64	DynamiQ port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	slave	Signal	CHI defined pins.
broadcastcachemaint	slave	Signal	ACE defined pins.
broadcastouter	slave	Signal	ACE defined pins.
broadcastpersist	slave	Signal	CHI defined pins.
cfgend	slave	Signal	This signal if for EE bit initialisation.
cfgte	slave	Signal	This signal provides default exception handling state.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
clusterid	slave	Value	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set ManagerID64 of the core.
clusterpmuirq	master	Signal	DynamiQ pmu irq
CNTHPIRQ	master	Signal	Timer signals to SOC.
CNTHVIRQ	master	Signal	Timer signals to SOC.
CNTPNSIRQ	master	Signal	Timer signals to SOC.
CNTPSIRQ	master	Signal	Timer signals to SOC.
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.
CNTVIRQ	master	Signal	Timer signals to SOC.
commirq	master	Signal	Interrupt signal from debug communications channel.

Port	Direction	Protocol	Description
core_clk_in	slave	ClockSignal	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
cpuporeset	slave	Signal	CPU power on reset. Initializes all the processor logic, including debug logic.
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
ctidbgirq	master	Signal	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	slave	Signal	External debug interface.
dbgnopwrdown	master	Signal	No power-down request.
dbgppwrupreq	master	Signal	Debug power up request.
dev_debug_s	slave	PVBus	External debug interface.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	slave	PVBus	External debug interface.
niden	slave	Signal	External debug interface.
pchannel_cluster	slave	PChannel	PChannel for cluster.
pchannel_core	slave	PChannel	PChannels for cores
pmbirq	master	Signal	Interrupt signal from statistical profiling unit.
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	master	PVBus	The core will generate bus requests on this port.
pvbus_periph_m	master	PVBus	The core can generate peripheral bus request on this port.
rei	slave	Signal	Per core RAM Error Interrupt.
reset	slave	Signal	Raising this signal will put the core into reset mode.
rvbaraddr	slave	Value_64	Reset vector base address.
sci_m	master	SystemCoherencyInterface	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei	slave	Signal	Per core System Error physical pins.
spiden	slave	Signal	External debug interface.
spniden	slave	Signal	External debug interface.
sporeset	slave	Signal	A single cluster-wide power on reset signal for all resettable registers in DynamIQ.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfiq	slave	Signal	Virtualised FIQ.

Port	Direction	Protocol	Description
vinithi	slave	Signal	This signal controls of the location of the exception vectors at reset.
virq	slave	Signal	Virtualised IRQ.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L3 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Per core virtual System Error physical pins.

## Parameters for ARMCortexA65AECT\_CortexA76AECT

### BROADCASTATOMIC

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`

Default value: `true`

### BROADCASTCACHEMAINT

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

Type: `bool`

Default value: `true`

### BROADCASTOUTER

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

Type: `bool`

Default value: `true`

### BROADCASTPERSIST

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: `bool`

Default value: `true`

### CLUSTER\_ID

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels,

the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the ManagerID64 of the core.

Type: `uint32_t`

Default value: `0x0`

### **CMO\_broadcast\_when\_cache\_state\_modelling\_disabled**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. This value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required.

Type: `uint8_t`

Default value: `1`

### **GICDISABLE**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`

Default value: `true`

### **dcache-state\_modelled**

Set whether D-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **default\_opmode**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

Type: `uint8_t`

Default value: `4`

### **diagnostics**

Enable DynamIQ diagnostic messages.

Type: `bool`

Default value: `false`

### **`enable_lock_step`**

Whether the core is configured in Dual Core Lock Step mode (FEAT\_DCLS).

Type: `bool`

Default value: `false`

### **`enable_simulation_performance_optimizations`**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of `stage12_tlb_size` parameter to 1024).

Type: `bool`

Default value: `true`

### **`has_acp`**

If true, Accelerator Coherency Port is configured.

Type: `bool`

Default value: `false`

### **`has_peripheral_port`**

If true, additional AXI peripheral port is configured.

Type: `bool`

Default value: `false`

### **`icache-state_modelled`**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **`l3cache-hit_latency`**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-maintenance\_latency**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-miss\_latency**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-read\_access\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size `l3cache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-read\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l3cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-size**

L3 Cache size in bytes.

Type: `uint32_t`

Default value: `0x100000`

**l3cache-snoop\_data\_transfer\_latency**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-snoop_issue_latency`**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-write_access_latency`**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size `l3cache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-write_latency`**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l3cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`pchannel_treat_simreset_as_poreset`**

Register core as ON state to cluster with simulation reset.

Type: `bool`

Default value: `false`

### **`periph_address_end`**

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

Type: `uint64_t`

Default value: `0x0`

### **`periph_address_start`**

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).



Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.NUM_CORES`**

Number of cores per cluster.

Type: `uint8_t`

Default value: `2`

### **`subcluster0.cpi_div`**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

### **`subcluster0.cpi_mul`**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

### **`subcluster0.cpu0.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu0.enable_single_thread_at_reset`**

Enable single thread after reset and keep other thread in reset.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu0.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu0.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x40000

**subcluster0.cpu0.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu0.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu0.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster0.cpu0.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster0.cpu0.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: 0x123456

**subcluster0.cpu0.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster0.cpu0.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster0.cpu0.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster0.cpu0.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu0.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu0.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu0.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu0.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu0.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu0.thread0.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian.

Type: `bool`

Default value: `false`

**subcluster0.cpu0.thread0.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster0.cpu0.thread0.MPIDR-override**

Override MPIDR value. A nonzero value will override the MT,cluster,cpu,thread ID bits in MPIDR.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu0.thread0.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu0.thread0.VINITHI**

Reset value of SCTLR.V.

Type: `bool`

Default value: `false`

**subcluster0.cpu0.thread1.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian.

Type: `bool`

Default value: `false`

**subcluster0.cpu0.thread1.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster0.cpu0.thread1.MPIDR-override**

Override MPIDR value. A nonzero value will override the MT,cluster,cpu,thread ID bits in MPIDR.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu0.thread1.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: 0x0

**subcluster0.cpu0.thread1.VINITHI**

Reset value of SCTLR.V.

Type: bool

Default value: false

**subcluster0.cpu0.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu0.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster0.cpu0.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: true

**subcluster0.cpu1.CRYPTODISABLE**

Disable cryptographic features.

Type: bool

Default value: false

**subcluster0.cpu1.enable\_single\_thread\_at\_reset**

Enable single thread after reset and keep other thread in reset.

Type: bool

Default value: false

**subcluster0.cpu1.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter trace\_special\_hlt\_imm16.

Type: bool

Default value: false

**subcluster0.cpu1.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t



Default value: 0x0

#### **subcluster0.cpu1.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x40000

#### **subcluster0.cpu1.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu1.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu1.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu1.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu1.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu1.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu1.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu1.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu1.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu1.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster0.cpu1.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster0.cpu1.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: `N/A`

**subcluster0.cpu1.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu1.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu1.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu1.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu1.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu1.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu1.thread0.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian.

Type: `bool`

Default value: `false`

**subcluster0.cpu1.thread0.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster0.cpu1.thread0.MPIDR-override**

Override MPIDR value. A nonzero value will override the MT,cluster,cpu,thread ID bits in MPIDR.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu1.thread0.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu1.thread0.VINITHI**

Reset value of SCTL.R.V.

Type: `bool`

Default value: `false`

**subcluster0.cpu1.thread1.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian.

Type: `bool`

Default value: `false`

**subcluster0.cpu1.thread1.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster0.cpu1.thread1.MPIDR-override**

Override MPIDR value. A nonzero value will override the MT,cluster,cpu,thread ID bits in MPIDR.

Type: `uint64_t`

Default value: 0x0

**subcluster0.cpu1.thread1.RVBARADDR**

Value of RVBAR\_ELx register.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu1.thread1.VINITHI**

Reset value of SCTLR.V.

Type: bool

Default value: false

**subcluster0.cpu1.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu1.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster0.cpu1.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: true

**subcluster0.cpu2.CRYPTODISABLE**

Disable cryptographic features.

Type: bool

Default value: false

**subcluster0.cpu2.enable\_single\_thread\_at\_reset**

Enable single thread after reset and keep other thread in reset.

Type: `bool`

Default value: `false`

**subcluster0.cpu2.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu2.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu2.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu2.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu2.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu2.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu2.l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x40000`

**subcluster0.cpu2.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu2.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu2.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu2.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster0.cpu2.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu2.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu2.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu2.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu2.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu2.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster0.cpu2.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.



Type: `uint8_t`

Default value: 171

### **`subcluster0.cpu2.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **`subcluster0.cpu2.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **`subcluster0.cpu2.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster0.cpu2.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu2.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster0.cpu2.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster0.cpu2.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

#### **`subcluster0.cpu2.thread0.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu2.thread0.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu2.thread0.MPIDR-override`**

Override MPIDR value. A nonzero value will override the MT,cluster,cpu,thread ID bits in MPIDR.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu2.thread0.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu2.thread0.VINITHI`**

Reset value of SCTLR.V.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu2.thread1.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu2.thread1.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu2.thread1.MPIDR-override`**

Override MPIDR value. A nonzero value will override the MT,cluster,cpu,thread ID bits in MPIDR.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu2.thread1.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu2.thread1.VINITHI`**

Reset value of SCTLR.V.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu2.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster0.cpu2.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu2.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster0.cpu3.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster0.cpu3.enable\_single\_thread\_at\_reset**

Enable single thread after reset and keep other thread in reset.

Type: `bool`

Default value: `false`

**subcluster0.cpu3.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu3.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu3.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu3.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu3.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x40000

**subcluster0.cpu3.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu3.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu3.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

### **`subcluster0.cpu3.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

### **`subcluster0.cpu3.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu3.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu3.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu3.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster0.cpu3.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster0.cpu3.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster0.cpu3.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster0.cpu3.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu3.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: 0x0

**subcluster0.cpu3.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: 0xf000000

**subcluster0.cpu3.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu3.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu3.thread0.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian.

Type: `bool`

Default value: `false`

**subcluster0.cpu3.thread0.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster0.cpu3.thread0.MPIDR-override**

Override MPIDR value. A nonzero value will override the MT,cluster,cpu,thread ID bits in MPIDR.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu3.thread0.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu3.thread0.VINITHI**

Reset value of SCTLR.V.

Type: `bool`

Default value: `false`



**subcluster0.cpu3.thread1.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian.

Type: `bool`

Default value: `false`

**subcluster0.cpu3.thread1.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster0.cpu3.thread1.MPIDR-override**

Override MPIDR value. A nonzero value will override the MT,cluster,cpu,thread ID bits in MPIDR.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu3.thread1.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu3.thread1.VINITHI**

Reset value of SCTLR.V.

Type: `bool`

Default value: `false`

**subcluster0.cpu3.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu3.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu3.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu4.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu4.enable_single_thread_at_reset`**

Enable single thread after reset and keep other thread in reset.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu4.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu4.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu4.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu4.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x40000

**subcluster0.cpu4.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu4.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu4.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu4.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu4.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu4.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster0.cpu4.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster0.cpu4.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu4.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu4.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu4.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu4.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu4.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu4.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu4.thread0.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian.

Type: `bool`

Default value: `false`

**subcluster0.cpu4.thread0.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster0.cpu4.thread0.MPIDR-override**

Override MPIDR value. A nonzero value will override the MT,cluster,cpu,thread ID bits in MPIDR.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu4.thread0.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu4.thread0.VINITHI**

Reset value of SCTLR.V.

Type: `bool`

Default value: `false`

**subcluster0.cpu4.thread1.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian.

Type: `bool`

Default value: `false`

**subcluster0.cpu4.thread1.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster0.cpu4.thread1.MPIDR-override**

Override MPIDR value. A nonzero value will override the MT,cluster,cpu,thread ID bits in MPIDR.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu4.thread1.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu4.thread1.VINITHI**

Reset value of SCTLR.V.

Type: `bool`

Default value: `false`

**subcluster0.cpu4.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster0.cpu4.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu4.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster0.cpu5.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu5.enable_single_thread_at_reset`**

Enable single thread after reset and keep other thread in reset.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu5.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu5.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`



**subcluster0.cpu5.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu5.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu5.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu5.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu5.l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x40000`

**subcluster0.cpu5.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster0.cpu5.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu5.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu5.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu5.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu5.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu5.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster0.cpu5.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster0.cpu5.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster0.cpu5.semihosting-cwd**

Base directory for semihosting file access.

Type: string

Default value: N/A

**subcluster0.cpu5.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: `true`

**subcluster0.cpu5.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu5.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu5.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu5.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu5.thread0.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian.

Type: `bool`

Default value: `false`

**subcluster0.cpu5.thread0.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster0.cpu5.thread0.MPIDR-override**

Override MPIDR value. A nonzero value will override the MT,cluster,cpu,thread ID bits in MPIDR.

Type: `uint64_t`

Default value: 0x0

**subcluster0.cpu5.thread0.RVBARADDR**

Value of RVBAR\_ELx register.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.thread0.VINITHI**

Reset value of SCTLR.V.

Type: bool

Default value: false

**subcluster0.cpu5.thread1.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian.

Type: bool

Default value: false

**subcluster0.cpu5.thread1.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: bool

Default value: false

**subcluster0.cpu5.thread1.MPIDR-override**

Override MPIDR value. A nonzero value will override the MT,cluster,cpu,thread ID bits in MPIDR.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.thread1.RVBARADDR**

Value of RVBAR\_ELx register.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.thread1.VINITHI**

Reset value of SCTLR.V.

Type: bool

Default value: `false`

#### **`subcluster0.cpu5.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster0.cpu5.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu5.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster0.dcache-hit_latency`**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.dcache-maintenance_latency`**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.dcache-miss_latency`**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **subcluster0.dcache-prefetch\_enabled**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

Type: bool

Default value: false

### **subcluster0.dcache-read\_access\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster0.dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster0.dcache-size**

L1 D-Cache size in bytes.

Type: uint32\_t

Default value: 0x10000

### **subcluster0.dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster0.dcache-write\_access\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be

used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.dcache-write_latency`**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.force_zero_PSTATE_PAN`**

Non-architecture parameter to force `PSTATE.PAN` to be 0.0: No effect. 1: `PSTATE.PAN` is always treated as 0. The parameter optimizes the performance of updating `PSTATE.PAN`.

Type: `bool`

Default value: `false`

#### **`subcluster0.has_delayed_dbgreg`**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

#### **`subcluster0.has_delayed_sysreg`**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

#### **`subcluster0.has_dot_product`**

Possible values of this parameter are:- 1, feature not implemented.- 2, feature is implemented.

Type: `uint8_t`

Default value: `2`

#### **`subcluster0.icache-hit_latency`**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.



Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.icache-maintenance_latency`**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.icache-miss_latency`**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.icache-prefetch_enabled`**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

### **`subcluster0.icache-read_access_latency`**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.icache-read_latency`**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.icache-size**

L1 I-Cache size in bytes.

Type: uint32\_t

Default value: 0x10000

**subcluster0.invalidate\_code\_cache\_on\_icache\_cmo**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: uint8\_t

Default value: 0

**subcluster0.ptw\_latency**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

**subcluster0.reported\_patch\_level**

Purely cosmetic patch level value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: int8\_t

Default value: -1

**subcluster0.reported\_revision\_number**

Purely cosmetic revision number value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: int8\_t

Default value: -1

**subcluster0.tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

**subcluster0.tlbi\_stall\_enabled**

If true, tlbi invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: `bool`

Default value: `false`

**subcluster0.walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.NUM\_CORES**

Number of cores per cluster.

Type: `uint8_t`

Default value: `2`

**subcluster1.cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**subcluster1.cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**subcluster1.cpu0.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu0.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu0.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu0.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu0.VINITHI`**

Reset value of SCTLR.V.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu0.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu0.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu0.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu0.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu0.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu0.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu0.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster1.cpu0.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu0.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu0.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu0.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu0.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu0.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu0.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu0.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu0.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu0.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster1.cpu0.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster1.cpu0.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu0.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu0.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu0.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu0.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu0.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu0.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu0.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu0.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu1.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu1.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.



Type: `bool`

Default value: `false`

### **`subcluster1.cpu1.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu1.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.VINITHI`**

Reset value of SCTLR.V.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu1.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu1.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu1.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu1.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu1.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu1.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu1.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster1.cpu1.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu1.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu1.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu1.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu1.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu1.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu1.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu1.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu1.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu1.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster1.cpu1.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster1.cpu1.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu1.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu1.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu1.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu1.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu1.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu1.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu1.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu1.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu2.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu2.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu2.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu2.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu2.VINITHI`**

Reset value of SCTLR.V.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu2.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu2.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu2.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu2.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu2.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu2.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster1.cpu2.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu2.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu2.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu2.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu2.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu2.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu2.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu2.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000



**subcluster1.cpu2.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu2.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster1.cpu2.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster1.cpu2.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu2.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu2.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu2.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu2.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu2.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu2.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu2.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu3.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu3.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu3.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu3.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu3.VINITHI`**

Reset value of SCTLR.V.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu3.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu3.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu3.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster1.cpu3.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu3.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu3.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu3.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu3.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu3.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster1.cpu3.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster1.cpu3.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu3.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu3.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu3.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu3.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu3.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu3.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu3.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.dcache-maintenance\_latency**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.dcache-miss\_latency**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.dcache-prefetch\_enabled**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

Type: bool

Default value: false

**subcluster1.dcache-read\_access\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.



Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.dcache-write_access_latency`**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.dcache-write_latency`**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.ext_abort_device_read_is_sync`**

Synchronous reporting of device-nGnRE read external aborts.

Type: `bool`

Default value: `false`

#### **`subcluster1.ext_abort_device_write_is_sync`**

Synchronous reporting of device-nGnRE write external aborts.

Type: `bool`

Default value: `false`

#### **`subcluster1.ext_abort_so_read_is_sync`**

Synchronous reporting of device-nGnRnE read external aborts.

Type: `bool`

Default value: `false`

#### **`subcluster1.ext_abort_so_write_is_sync`**

Synchronous reporting of device-nGnRnE write external aborts.

Type: `bool`

Default value: `false`

#### **`subcluster1.force_zero_PSTATE_PAN`**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

#### **`subcluster1.has_delayed_dbgreg`**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

#### **`subcluster1.has_delayed_sysreg`**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

#### **`subcluster1.has_dot_product`**

Possible values of this parameter are:- 1, feature not implemented.- 2, feature is implemented.

Type: `uint8_t`

Default value: 2

#### **`subcluster1.has_statistical_profiling`**

Whether Statistical Based Profiling is implemented (FEAT\_SPE).

Type: `bool`

Default value: `true`

#### **`subcluster1.icache-hit_latency`**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

**subcluster1.icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.invalidate\_code\_cache\_on\_icache\_cmo**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: `uint8_t`

Default value: 0

**subcluster1.ptw\_latency**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

**subcluster1.reported\_patch\_level**

Purely cosmetic patch level value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: `int8_t`

Default value: -1

**subcluster1.reported\_revision\_number**

Purely cosmetic revision number value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: `int8_t`

Default value: -1

**subcluster1.tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

**subcluster1.tlbi\_stall\_enabled**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: `bool`

Default value: `false`

**subcluster1.treat-dcache-cmos-to-pou-as-nop**

Whether dcache invalidation to the point of unification is required for instruction to data coherence. 0 - Invalidate ops required, 1 - Invalidate ops not required and cannot generate faults, 2 - Invalidate ops not required but can generate faults.

Type: uint8\_t

Default value: 0

**subcluster1.walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

## 3.63 ARMCortexA65CT

Defined in `LISA/ARMCortexA65CT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following parameters were added:

- `has_delayed_dbgreg`
- `has_delayed_sysreg`

### About ARMCortexA65CT

The model supports the following features:

- DynamIQ (DSU) system registers.
- Per-core L2 cache.
- A P-Channel for the cluster and for each core.
- Optional peripheral port.
- L3 cache partition.
- Per-core clock.
- `BROADCASTPERSIST` pin.

The model does not support the following features:

- BROADCASTCACHEMAINTPOU pin.
- COREINSTRRET, COREINSTRRUN, or nPMBIRQ signals.
- DynamIQ features that are irrelevant to the programmers' view simulation, for example:
  - Automatic CPU retention mode.
  - Level-3 Cache RAM retention.
- 256-bit wide output transactions.
- ECC and parity schemes are hardware-specific so are not supported.
- Self-test features (MBIST).
- Latency configuration.
- Snoop filtering.
- Debug power management signals DBGWRDUP and DBGRSTREQ are not implemented, but DBGWRUPREQ and DBGNOPWRDWN are implemented.
- Cache stashing capability.
- Each thread has its own L1Cache and L2Cache instead of a common cache that is shared by all threads of the core.
- Per-thread parameters are not yet supported but signals are implemented.

This model has a variable number of cores per cluster, specified using the NUM\_CORES parameter.

The per-core parameters are preceded by `cpun.`, where n identifies the core (0-3).

This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (`crypto.dll` or `crypto.so`) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

## Iris and MTI instances for ARMCortexA65CT

This model has the following Iris instances:

Name	Instance type
ARMCortexA65CT	Cluster_ARM_Cortex-A65
ARMCortexA65CT.AMU	PVBusLogger
ARMCortexA65CT.AMU.mapper	PVBusMapper
ARMCortexA65CT.DAP	PVBusLogger
ARMCortexA65CT.DAP.mapper	PVBusMapper
ARMCortexA65CT.DSU	DSU
ARMCortexA65CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA65CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA65CT.DSU.shared_cache	PVCache
ARMCortexA65CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA65CT.DSU.shared_cache.upstream[Y] (where Y = 0-3)	PVBusSlave

Name	Instance type
ARMCortexA65CT.MMAP	PVBusLogger
ARMCortexA65CT.MMAP.mapper	PVBusMapper
ARMCortexA65CT.RAS	PVBusLogger
ARMCortexA65CT.RAS.mapper	PVBusMapper
ARMCortexA65CT.cpu0.dtlb	TLB
ARMCortexA65CT.cpu0.threadZ (where Z = 0-1)	ARM_Cortex-A65
ARMCortexA65CT.cpu0.threadZ.UTLB (where Z = 0-1)	TLB
ARMCortexA65CT.cpu0.thread0.l1dcache	PVCache
ARMCortexA65CT.cpu0.thread0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA65CT.cpu0.thread0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA65CT.cpu0.thread0.l1icache	PVCache
ARMCortexA65CT.cpu0.thread0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA65CT.cpu0.thread0.l1icache.upstream[0]	PVBusSlave
ARMCortexA65CT.cpu0.thread0.l2cache	PVCache
ARMCortexA65CT.cpu0.thread0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA65CT.cpu0.thread0.l2cache.upstream[V] (where V = 0-1)	PVBusSlave
ARMCortexA65CT.ext_bus	PVBusLogger
ARMCortexA65CT.ext_bus.mapper	PVBusMapper
ARMCortexA65CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA65CT.global_debug_rom	debug_rom

This model has the following MTI trace components:

Name	Component type
ARMCortexA65CT.AMU	PVBusLogger
ARMCortexA65CT.AMU.mapper	PVBusMapper
ARMCortexA65CT.DAP	PVBusLogger
ARMCortexA65CT.DAP.mapper	PVBusMapper
ARMCortexA65CT.DSU	DSU
ARMCortexA65CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA65CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA65CT.DSU.shared_cache	PVCache
ARMCortexA65CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA65CT.DSU.shared_cache.upstream[Y] (where Y = 0-3)	PVBusSlave
ARMCortexA65CT.MMAP	PVBusLogger
ARMCortexA65CT.MMAP.mapper	PVBusMapper
ARMCortexA65CT.RAS	PVBusLogger
ARMCortexA65CT.RAS.mapper	PVBusMapper
ARMCortexA65CT.cpu0.threadZ (where Z = 0-1)	ARM_Cortex-A65
ARMCortexA65CT.cpu0.threadZ.UTLB (where Z = 0-1)	TLB
ARMCortexA65CT.cpu0.thread0.l1dcache	PVCache

Name	Component type
ARMCortexA65CT.cpu0.thread0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA65CT.cpu0.thread0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA65CT.cpu0.thread0.l1licache	PVCache
ARMCortexA65CT.cpu0.thread0.l1licache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA65CT.cpu0.thread0.l1licache.upstream[0]	PVBusSlave
ARMCortexA65CT.cpu0.thread0.l2cache	PVCache
ARMCortexA65CT.cpu0.thread0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA65CT.cpu0.thread0.l2cache.upstream[V] (where V = 0-1)	PVBusSlave
ARMCortexA65CT.ext_bus	PVBusLogger
ARMCortexA65CT.ext_bus.mapper	PVBusMapper
ARMCortexA65CT.gic_cpuif_decoder_cluster	GLCv3CPUInterfaceDecoder

## Ports for ARMCortexA65CT

Port	Direction	Protocol	Description
acp_s	slave	PVBus	AXI ACP slave port.
AENDMP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
ASTARTMP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	slave	Signal	CHI defined pins.
broadcastcachemaint	slave	Signal	ACE defined pins.
broadcastouter	slave	Signal	ACE defined pins.
broadcastpersist	slave	Signal	CHI defined pins.
cfgend	slave	Signal	This signal if for EE bit initialisation.
cfgte	slave	Signal	This signal provides default exception handling state.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
clusterid	slave	Value	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set ManagerID64 of the core.
clusterpmuirq	master	Signal	DynamiQ pmu irq
CNTHPIRQ	master	Signal	Timer signals to SOC.
CNTHVIRQ	master	Signal	Timer signals to SOC.
CNTPNSIRQ	master	Signal	Timer signals to SOC.
CNTPSIRQ	master	Signal	Timer signals to SOC.



Port	Direction	Protocol	Description
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.
CNTVIRQ	master	Signal	Timer signals to SOC.
commirq	master	Signal	Interrupt signal from debug communications channel.
core_clk_in	slave	ClockSignal	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
cpuporeset	slave	Signal	Power on reset. Initializes all the processor logic, including debug logic.
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
ctidbgirq	master	Signal	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	slave	Signal	External debug interface.
dbgnopwrdown	master	Signal	These signals relate to core power down.
dbgprupreq	master	Signal	Debug power up request.
dev_debug_s	slave	PVBus	External debug interface.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	slave	PVBus	External debug interface.
niden	slave	Signal	External debug interface.
pchannel_cluster	slave	PChannel	PChannel for cluster.
pchannel_core	slave	PChannel	PChannels for cores
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	master	PVBus	The core will generate bus requests on this port.
pvbus_periph_m	master	PVBus	The core can generate peripheral bus request on this port.
rei	slave	Signal	Per core RAM Error Interrupt.
reset	slave	Signal	Raising this signal will put the core into reset mode.
rvbaraddr	slave	Value_64	Reset vector base address.
sci_m	master	SystemCoherencyInterface	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei	slave	Signal	Per core virtual System Error physical pins.
spiden	slave	Signal	External debug interface.
spniden	slave	Signal	External debug interface.
sporeset	slave	Signal	A single cluster-wide power on reset signal for all resettable registers in DynamIQ.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.

Port	Direction	Protocol	Description
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfiq	slave	Signal	Virtualised FIQ.
vinithi	slave	Signal	This signal controls of the location of the exception vectors at reset.
virq	slave	Signal	Virtualised IRQ.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Per core virtual System Error physical pins.

## Parameters for ARM Cortex A65CT

### **cpuX.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **cpuX.enable\_single\_thread\_at\_reset**

Enable single thread after reset and keep other thread in reset.

Type: `bool`

Default value: `false`

### **cpuX.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **cpuX.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **cpuX.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **cpuX.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **cpuX.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **cpuX.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **cpuX.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x40000

### **cpuX.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **cpuX.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

### **`cpuX.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

### **`cpuX.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`cpuX.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**cpuX.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**cpuX.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**cpuX.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**cpuX.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**cpuX.semihosting-cwd**

Base directory for semihosting file access.

Type: string

Default value: N/A

**cpuX.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: true

**cpuX.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**cpuX.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**cpuX.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**cpuX.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**cpuX.thread0.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian.

Type: bool

Default value: false

**cpuX.thread0.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: bool

Default value: false

**cpuX.thread0.MPIDR-override**

Override MPIDR value. A nonzero value will override the MT,cluster,cpu,thread ID bits in MPIDR.

Type: uint64\_t

Default value: 0x0

**cpuX.thread0.RVBARADDR**

Value of RVBAR\_ELx register.

Type: uint64\_t

Default value: 0x0

**cpuX.thread0.VINITHI**

Reset value of SCTLR.V.

Type: bool

Default value: false

**cpuX.thread1.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian.

Type: bool

Default value: false

**cpuX.thread1.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: bool

Default value: false

**cpuX.thread1.MPIDR-override**

Override MPIDR value. A nonzero value will override the MT,cluster,cpu,thread ID bits in MPIDR.

Type: uint64\_t

Default value: 0x0

**cpuX.thread1.RVBARADDR**

Value of RVBAR\_ELx register.

Type: uint64\_t

Default value: 0x0

**cpuX.thread1.VINITHI**

Reset value of SCTLR.V.

Type: bool

Default value: false

**cpuX.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`cpuX.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`cpuX.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **BROADCASTATOMIC**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`

Default value: `true`

### **BROADCASTCACHEMAINT**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

Type: `bool`

Default value: `true`

### **BROADCASTOUTER**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

Type: `bool`

Default value: `true`

### **BROADCASTPERSIST**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.



Type: `bool`

Default value: `true`

### **CLUSTER\_ID**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the ManagerID64 of the core.

Type: `uint32_t`

Default value: `0x0`

### **CMO\_broadcast\_when\_cache\_state\_modelling\_disabled**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. This value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain even when cache state modelling is disabled. Possible values are:- 0 = All CMOs are broadcast if architecturally required- 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required.

Type: `uint8_t`

Default value: `1`

### **GICDISABLE**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`

Default value: `true`

### **NUM\_CORES**

Number of cores per cluster.

Type: `uint8_t`

Default value: `1`

### **cluster\_patch\_level**

Cosmetic change to patch number in CLUSTERIDR. Corresponds to the Y in rXpY.

Type: `uint8_t`

Default value: 0

**cluster\_revision\_number**

Cosmetic change to revision number in CLUSTERIDR, Corresponds to the X in rXpY.

Type: uint8\_t

Default value: 0

**cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: uint32\_t

Default value: 0x1

**cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: uint32\_t

Default value: 0x1

**dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**dcache-maintenance\_latency**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**dcache-miss\_latency**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**dcache-prefetch\_enabled**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

Type: `bool`

Default value: `false`

**dcache-read\_access\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**dcache-size**

L1 D-Cache size in bytes.

Type: `uint32_t`

Default value: `0x10000`

**dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**dcache-state\_modelled**

Set whether D-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **`dcache-write_access_latency`**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`dcache-write_latency`**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`default_opmode`**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

Type: `uint8_t`

Default value: 4

### **`diagnostics`**

Enable DynamIQ diagnostic messages.

Type: `bool`

Default value: `false`

### **`enable_simulation_performance_optimizations`**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of `stage12_tlb_size` parameter to 1024).

Type: `bool`

Default value: `true`

**force\_zero\_PSTATE\_PAN**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

**has\_acp**

If true, Accelerator Coherency Port is configured.

Type: `bool`

Default value: `false`

**has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

**has\_dot\_product**

Possible values of this parameter are:- 1, feature not implemented.- 2, feature is implemented.

Type: `uint8_t`

Default value: 2

**has\_peripheral\_port**

If true, additional AXI peripheral port is configured.

Type: `bool`

Default value: `false`

**icache-hit\_latency**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`icache-maintenance_latency`**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`icache-miss_latency`**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`icache-prefetch_enabled`**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

### **`icache-read_access_latency`**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`icache-read_latency`**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-size**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: `0x10000`

**icache-state\_modelled**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

**invalidate\_code\_cache\_on\_icache\_cmo**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: `uint8_t`

Default value: `0`

**l3cache-hit\_latency**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-maintenance\_latency**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-miss\_latency**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-read\_access\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size l3cache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**l3cache-read\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. l3cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**l3cache-size**

L3 Cache size in bytes.

Type: uint32\_t

Default value: 0x400000

**l3cache-snoop\_data\_transfer\_latency**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**l3cache-snoop\_issue\_latency**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**l3cache-write\_access\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size l3cache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used



instead of per-byte even if l3cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **l3cache-write\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. l3cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **pchannel\_treat\_simreset\_as\_poreset**

Register core as ON state to cluster with simulation reset.

Type: bool

Default value: false

### **periph\_address\_end**

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

Type: uint64\_t

Default value: 0x0

### **periph\_address\_start**

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

Type: uint64\_t

Default value: 0x0

### **ptw\_latency**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

### **tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

### **tlbi\_stall\_enabled**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: bool

Default value: false

### **walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

## 3.64 ARMCortexA710CT

Defined in `LISA/ARMCortexA710CT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r2p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following parameters were added:

- `has_delayed_dbgreg`
- `has_delayed_sysreg`
- `num_acp`

### About ARMCortexA710CT

The model supports the following features:

- DynamIQ Shared Unit-110 (DSU-110) system registers.
- L2 cache is supported at the per-core level only and there is no implementation of Core-Complex with shared L2 cache yet.
- PChannel for the cluster and for each core.
- `BROADCASTPERSIST` pin.
- Optional peripheral port.

- L3Cache partition.
- Per-core clock.
- Utility bus.
- AArch32 at ELO.

Support for the following features is planned for a future release:

- BROADCASTCACHEMAINTPOU pin.
- COREINSTRRET, COREINSTRRUN, and nPMBIRQ signals.

The following features will not be implemented:

- DynamIQ features that are irrelevant to the programmers' view simulation, for example:
  - Automatic CPU retention mode.
  - Level-3 Cache RAM retention.
- 256-bit wide output transactions.
- Error correction or detection features.
- Self-test features (MBIST).
- Snoop filtering.

This model supports the Arm(R)v8-A Cryptographic Extensions, which requires the crypto plug-in (crypto.dll or crypto.so) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

## Iris and MTI instances for ARM Cortex-A710CT

This model has the following Iris instances:

Name	Instance type
ARM Cortex-A710CT	Cluster_ARM_Cortex-A710
ARM Cortex-A710CT.AMU	PVBusLogger
ARM Cortex-A710CT.AMU.mapper	PVBusMapper
ARM Cortex-A710CT.DAP	PVBusLogger
ARM Cortex-A710CT.DAP.mapper	PVBusMapper
ARM Cortex-A710CT.DSU	DSU-110
ARM Cortex-A710CT.DSU.PPU_cluster	PPUv1
ARM Cortex-A710CT.DSU.PPU_cluster.busslave	PVBusSlave
ARM Cortex-A710CT.DSU.PPU_core0	PPUv1
ARM Cortex-A710CT.DSU.PPU_core0.busslave	PVBusSlave
ARM Cortex-A710CT.DSU.mpam_busslave	PVBusSlave
ARM Cortex-A710CT.DSU.shared_cache	PVCache
ARM Cortex-A710CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARM Cortex-A710CT.DSU.shared_cache.upstream[Y] (where Y = 0-3)	PVBusSlave
ARM Cortex-A710CT.DSU.utility_slave[0]	PVBusSlave

Name	Instance type
ARMCortexA710CT.MMAP	PVBusLogger
ARMCortexA710CT.MMAP.mapper	PVBusMapper
ARMCortexA710CT.RAS	PVBusLogger
ARMCortexA710CT.RAS.mapper	PVBusMapper
ARMCortexA710CT.cpu0	ARM_Cortex-A710
ARMCortexA710CT.cpu0.UTLB	TLB
ARMCortexA710CT.cpu0.debug_rom	debug_rom
ARMCortexA710CT.cpu0.dtlb	TLB
ARMCortexA710CT.cpu0.l1dcache	PVCache
ARMCortexA710CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA710CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA710CT.cpu0.l1icache	PVCache
ARMCortexA710CT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA710CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA710CT.cpu0.l2cache	PVCache
ARMCortexA710CT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA710CT.cpu0.l2cache.upstream[U] (where U = 0-1)	PVBusSlave
ARMCortexA710CT.ext_bus	PVBusLogger
ARMCortexA710CT.ext_bus.mapper	PVBusMapper
ARMCortexA710CT.gic_cpuif_decoder_cluster	GLICv3CPUInterfaceDecoder
ARMCortexA710CT.global_debug_rom	debug_rom
ARMCortexA710CT.secondary_debug_rom	debug_rom
ARMCortexA710CT.sve	ScalableVectorExtension

This model has the following MTI trace components:

Name	Component type
ARMCortexA710CT.AMU	PVBusLogger
ARMCortexA710CT.AMU.mapper	PVBusMapper
ARMCortexA710CT.DAP	PVBusLogger
ARMCortexA710CT.DAP.mapper	PVBusMapper
ARMCortexA710CT.DSU	DSU-110
ARMCortexA710CT.DSU.PPU_cluster	PPUv1
ARMCortexA710CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA710CT.DSU.PPU_core0	PPUv1
ARMCortexA710CT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexA710CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA710CT.DSU.shared_cache	PVCache
ARMCortexA710CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA710CT.DSU.shared_cache.upstream[Y] (where Y = 0-3)	PVBusSlave
ARMCortexA710CT.DSU.utility_slave[0]	PVBusSlave

Name	Component type
ARMCortexA710CT.MMAP	PVBusLogger
ARMCortexA710CT.MMAP.mapper	PVBusMapper
ARMCortexA710CT.RAS	PVBusLogger
ARMCortexA710CT.RAS.mapper	PVBusMapper
ARMCortexA710CT.cpu0	ARM_Cortex-A710
ARMCortexA710CT.cpu0.UTLB	TLB
ARMCortexA710CT.cpu0.l1dcache	PVCache
ARMCortexA710CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA710CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA710CT.cpu0.l1icache	PVCache
ARMCortexA710CT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA710CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA710CT.cpu0.l2cache	PVCache
ARMCortexA710CT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA710CT.cpu0.l2cache.upstream[U] (where U = 0-1)	PVBusSlave
ARMCortexA710CT.ext_bus	PVBusLogger
ARMCortexA710CT.ext_bus.mapper	PVBusMapper
ARMCortexA710CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

## Ports for ARMCortexA710CT

Port	Direction	Protocol	Description
acp_s	slave	PVBus	AXI ACP slave port.
AEND0MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND2MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND3MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART2MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	slave	Signal	CHI defined pins.
broadcastcachemaint	slave	Signal	ACE defined pins.
broadcastouter	slave	Signal	ACE defined pins.
broadcastpersist	slave	Signal	CHI defined pins.
cfgend	slave	Signal	This signal is for EE bit initialisation.

Port	Direction	Protocol	Description
cfgsdisable	slave	Signal	This signal disables write access to some secure Interrupt Controller registers.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcs_m_pchannel	master	PChannel	Cluster PCSM signal
cluster_powerdown_out	master	Signal	DEPRECATED - An output from PPU that informs thermal controller of the cluster power info
cluster_ppu_hw_stat	master	Value	Cluster PPU power state
clusterid	slave	Value	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set ManagerID64 of the core.
clusterpmuirq	master	Signal	DynamiQ pmu irq
CNTHPIRQ	master	Signal	Timer signals to SOC.
CNTHPSIRQ	master	Signal	Timer signals to SOC
CNTHVIRQ	master	Signal	Timer signals to SOC.
CNTHVSIRQ	master	Signal	Timer signals to SOC
CNTPNSIRQ	master	Signal	Timer signals to SOC.
CNTPSIRQ	master	Signal	Timer signals to SOC.
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.
CNTVIRQ	master	Signal	Timer signals to SOC.
commirq	master	Signal	Interrupt signal from debug communications channel.
core_clk_in	slave	ClockSignal	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
core_pcs_m_pchannel	master	PChannel	Core PCSM signals
core_powerdown_out	master	Signal	DEPRECATED - An output from PPU that informs thermal controller of the core power info
core_ppu_hw_stat	master	Value	Notify the power state change inside the PPU.
coreerrirq	master	Signal	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq	master	Signal	Fault indicator for a detected 1 or 2 bit ECC error
coreinstrrun	master	Signal	Core Running State
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
ctidbgirq	master	Signal	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	slave	Signal	External debug interface.

Port	Direction	Protocol	Description
dbgnopwrdown	master	Signal	No power-down request.
dbgpwrupreq	master	Signal	Debug power up request.
defaultMP	slave	Signal	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main master interface ports.
dev_debug_s	slave	PVBus	External debug interface.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.
gicreset	master	Signal	An output from PPU that can be used to reset GICCLK domain components
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	slave	PVBus	External debug interface.
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	master	Signal	PPU cluster interrupt.
ppu_cluster_isolate	master	Signal	PPU Cluster Domain isolation control.
ppu_cluster_wakerequest	slave	Signal	PPU cluster wake request signal.
ppu_core_irq	master	Signal	PPU core interrupt.
ppu_core_isolate	master	Signal	PPU Core Domain isolation control.
ppu_core_wakerequest	slave	Signal	PPU Core wake request signals.
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	master	PVBus	The core will generate bus requests on this port.
pvbus_periph_m	master	PVBus	The core can generate peripheral bus request on this port.
rei	slave	Signal	Per core RAM Error Interrupt.
reset	slave	Signal	Reset signal to cluster.
rvbaraddr	slave	Value_64	Reset vector base address.
sci_m	master	SystemCoherencyInterface	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei	slave	Signal	Per core System Error physical pins.
spiden	slave	Signal	External debug interface.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq	master	Signal	Interrupt signal from the trace buffer unit.
utility_bus_s	slave	PVBus	Utility bus slave
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfiq	slave	Signal	Virtualised FIQ.
virq	slave	Signal	Virtualised IRQ.

Port	Direction	Protocol	Description
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Per core virtual System Error physical pins.

## Parameters for ARM Cortex A710CT

### **cpuX.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **cpuX.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **cpuX.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **cpuX.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **cpuX.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`



**cpuX.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**cpuX.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**cpuX.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**cpuX.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**cpuX.l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

**cpuX.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**cpuX.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**cpuX.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**cpuX.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**cpuX.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**cpuX.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**cpuX.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**cpuX.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**cpuX.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**cpuX.semihosting-cwd**

Base directory for semihosting file access.

Type: string

Default value: N/A

**cpuX.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: `true`

**cpuX.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**cpuX.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**cpuX.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**cpuX.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**cpuX.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**cpuX.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**AEND0\_DEFAULT**

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

Type: `uint64_t`

Default value: `0x0`

**AEND1\_DEFAULT**

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

Type: `uint64_t`

Default value: `0x0`

**AEND2\_DEFAULT**

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

Type: `uint64_t`

Default value: `0x0`

**AEND3\_DEFAULT**

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART0\_DEFAULT**

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART1\_DEFAULT**

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART2\_DEFAULT**

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART3\_DEFAULT**

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

Type: `uint64_t`

Default value: `0x0`

**BROADCASTATOMIC**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`

Default value: `true`

**BROADCASTCACHEMAINT**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

Type: `bool`

Default value: `true`

**BROADCASTOUTER**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

Type: `bool`

Default value: `true`

**BROADCASTPERSIST**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: `bool`

Default value: `true`

**CLUSTER\_ID**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the ManagerID64 of the core.

Type: `uint32_t`

Default value: `0x0`

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. This value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required.

Type: `uint8_t`

Default value: `1`

**GICDISABLE**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`

Default value: `true`

**NUM\_CORES**

Number of cores per cluster.

Type: `uint8_t`

Default value: `1`

**core\_power\_on\_by\_default**

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPUs which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

Type: `bool`

Default value: `false`

**cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-maintenance\_latency**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-miss\_latency**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-prefetch\_enabled**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

Type: `bool`

Default value: `false`



**dcache-read\_access\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**dcache-size**

L1 D-Cache size in bytes.

Type: uint32\_t

Default value: 0x8000

**dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**dcache-state\_modelled**

Set whether D-cache has stateful implementation.

Type: bool

Default value: false

**dcache-write\_access\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be

used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`dcache-write_latency`**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`default_opmode`**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

Type: `uint8_t`

Default value: 4

### **`diagnostics`**

Enable DynamIQ diagnostic messages.

Type: `bool`

Default value: `false`

### **`enable_simulation_performance_optimizations`**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of `stage12_tlb_size` parameter to 1024).

Type: `bool`

Default value: `true`

### **`ete.CLAIMTAGS`**

Number of claim tags.

Type: `uint8_t`

Default value: 32

**ete.MAX\_INST\_PER\_Q**

Maximum limit for the number of instructions implied by a Q element.

Type: `uint16_t`

Default value: `0x1`

**ete.NumberOfRSPairs**

Number of resource selector pairs.

Type: `uint8_t`

Default value: `8`

**ete.PIDR\_CM0D**

TRCPIDR CM0D value.

Type: `uint8_t`

Default value: `0`

**ete.PIDR\_REVAND**

TRCPIDR REVAND value.

Type: `uint8_t`

Default value: `1`

**ete.PIDR\_REVISION**

TRCPIDR REVISION value.

Type: `uint8_t`

Default value: `0`

**ete.Q\_CADENCE**

Number of instruction blocks traced between two Q elements.

Type: `uint16_t`

Default value: `0x1`

**ete.RES0\_STATEFUL**

Whether **RES0** bits are stateful or **RAZ/WI**.

Type: `bool`

Default value: `false`

**ete.RETSTACK**

Return stack depth.

Type: `uint8_t`

Default value: 3

**ete.REVISION**

TRCIDR1 revision value.

Type: `uint8_t`

Default value: 2

**ete.SIM\_OVERFLOW\_GRANULARITY**

Number of instruction blocks in each granule, for simulated overflow.

Type: `uint32_t`

Default value: 0x64

**ete.SIM\_OVERFLOW\_PERCENTAGE**

Percentage of instruction blocks lost in each granule, for simulated overflow.

Type: `uint8_t`

Default value: 0

**ete.SOURCE\_ADDRESS**

Allow generation of source address elements.

Type: `bool`

Default value: `false`

**ete.TRACE\_OUTPUT**

File to which to write trace byte stream.

Type: `string`

Default value: `N/A`

**ete.TRCSRSTA\_FORCED\_EXCEP**

TRCSR.TA value for a forcibly traced exception.

Type: `bool`

Default value: `false`

**ext\_abort\_device\_read\_is\_sync**

Synchronous reporting of device-nGnRE read external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_device\_write\_is\_sync**

Synchronous reporting of device-nGnRE write external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_so\_read\_is\_sync**

Synchronous reporting of device-nGnRnE read external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_so\_write\_is\_sync**

Synchronous reporting of device-nGnRnE write external aborts.

Type: `bool`

Default value: `false`

**force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks**

Force MTE tag accesses to **RAZ/WI** and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

Type: `bool`

Default value: `false`

**force\_zero\_PSTATE\_PAN**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

**force\_zero\_mpam\_partid\_and\_pmg**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

Type: `uint8_t`

Default value: `0`

**has\_acp**

If true, Accelerator Coherency Port is configured.

Type: `bool`

Default value: `false`

**has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**has\_ete**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (`-plugin` or `-P`).

Type: `bool`

Default value: `false`

**has\_peripheral\_port**

If true, an additional AXI peripheral port is configured.

Type: `bool`

Default value: `false`

**icache-hit\_latency**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

**icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to

the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`icache-size`**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: `0x8000`

### **`icache-state_modelled`**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **`invalidate_code_cache_on_icache_cmo`**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: `uint8_t`

Default value: `0`

### **`l3cache-hit_latency`**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-maintenance_latency`**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-miss_latency`**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.



Type: `uint64_t`

Default value: `0x0`

### **`l3cache-read_access_latency`**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size `l3cache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-read_latency`**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l3cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-size`**

L3 Cache size in bytes.

Type: `uint32_t`

Default value: `0x0`

### **`l3cache-snoop_data_transfer_latency`**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-snoop_issue_latency`**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-write\_access\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size l3cache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**l3cache-write\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. l3cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**memory\_tagging\_support\_level**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).

Type: uint8\_t

Default value: 2

**mpmm\_accumulator\_multiplier**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator.value of n means the accumulator will use (n \* accumulator value) to calculate the mpmm threshold (MPMM).is provided as a fast model workaround to handle cases where execution of quantums in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

Type: uint8\_t

Default value: 1

**num\_acp**

Number of ACP ports.

Type: uint8\_t

Default value: 0

**num\_nodes**

Number of transport nodes. Zero implies direct-connect configuration.

Type: `uint8_t`

Default value: 1

### **`periph_address_end`**

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

Type: `uint64_t`

Default value: 0x0

### **`periph_address_start`**

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

Type: `uint64_t`

Default value: 0x0

### **`ptw_latency`**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

### **`tlb_latency`**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

### **`tlbi_stall_enabled`**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: `bool`

Default value: `false`

### **`treat-dcache-cmos-to-pou-as-nop`**

Whether dcache invalidation to the point of unification is required for instruction to data coherence. 0 - Invalidate ops required, 1 - Invalidate ops not required and cannot generate faults, 2 - Invalidate ops not required but can generate faults.

Type: `uint8_t`

Default value: 0

**walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

3.65 ARMCortexA715CT

Defined in LISA/ARMCortexA715CT.lisa.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r1p2	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

Changes in 11.31.15

The following parameters were added:

- has\_delayed\_dbgreg
- has\_delayed\_sysreg
- num\_acp

About ARMCortexA715CT

The model supports the following features:

- DynamIQ (DSU) system registers.
- DynamIQ r3p0.
- L2 cache is supported at the per-core level only and there is no implementation of Core-Complex with shared L2 cache yet.
- P-Channel for the cluster and for each core.
- Optional peripheral port.
- L3 cache partition.
- Per-core clock.
- BROADCASTPERSIST pin.
- Utility bus.
- Support for Interrupt signals from SPE is added as pmbirq[8].

Support for the following features is planned for a future release:

- TRBE.

- Core-Complex.
- BROADCASTCACHEMAINTPOU pin.
- COREINSTRRET, COREINSTRRUN, or nPMBIRQ signals.
- Each thread has its own L1Cache and L2Cache instead of a common cache that is shared by all threads of the core.
- Per-thread parameters are not yet supported but signals are implemented.

**Note**

The `cfgsdisable` signal will be removed in a future release.

The following features will not be implemented:

- DynamIQ features that are irrelevant to the programmers' view simulation, for example:
  - Automatic CPU retention mode.
  - Level-3 Cache RAM retention.
- 256-bit wide output transactions.
- ECC and parity schemes are hardware-specific so are not supported.
- Self-test features (MBIST).
- Latency configuration.
- Snoop filtering.
- Cache stashing capability.
- Embedded Logic Analyzer (ELA).

This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (`crypto.dll` or `crypto.so`) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

## Limitations

- Some of the SVE instructions do not raise undefined exceptions when SP is used as a source register.

## Iris and MTI instances for ARM Cortex-A715CT

This model has the following Iris instances:

Name	Instance type
ARMCortexA715CT	Cluster_ARM_Cortex-A715
ARMCortexA715CT.AMU	PVBusLogger
ARMCortexA715CT.AMU.mapper	PVBusMapper
ARMCortexA715CT.DAP	PVBusLogger

Name	Instance type
ARMCortexA715CT.DAP.mapper	PVBusMapper
ARMCortexA715CT.DSU	DSU-110
ARMCortexA715CT.DSU.PPU_cluster	PPUv1
ARMCortexA715CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA715CT.DSU.PPU_core0	PPUv1
ARMCortexA715CT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexA715CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA715CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA715CT.DSU.shared_cache	PVCache
ARMCortexA715CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA715CT.DSU.shared_cache.upstream[Y] (where Y = 0-3)	PVBusSlave
ARMCortexA715CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA715CT.MMAP	PVBusLogger
ARMCortexA715CT.MMAP.mapper	PVBusMapper
ARMCortexA715CT.RAS	PVBusLogger
ARMCortexA715CT.RAS.mapper	PVBusMapper
ARMCortexA715CT.cpu0	ARM_Cortex-A715
ARMCortexA715CT.cpu0.UTLB	TLB
ARMCortexA715CT.cpu0.debug_rom	debug_rom
ARMCortexA715CT.cpu0.dtlb	TLB
ARMCortexA715CT.cpu0.l1dcache	PVCache
ARMCortexA715CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA715CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA715CT.cpu0.l1icache	PVCache
ARMCortexA715CT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA715CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA715CT.cpu0.l2cache	PVCache
ARMCortexA715CT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA715CT.cpu0.l2cache.upstream[U] (where U = 0-1)	PVBusSlave
ARMCortexA715CT.ext_bus	PVBusLogger
ARMCortexA715CT.ext_bus.mapper	PVBusMapper
ARMCortexA715CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA715CT.global_debug_rom	debug_rom
ARMCortexA715CT.secondary_debug_rom	debug_rom
ARMCortexA715CT.sve	ScalableVectorExtension

This model has the following MTI trace components:

Name	Component type
ARMCortexA715CT.AMU	PVBusLogger
ARMCortexA715CT.AMU.mapper	PVBusMapper

Name	Component type
ARMCortexA715CT.DAP	PVBusLogger
ARMCortexA715CT.DAP.mapper	PVBusMapper
ARMCortexA715CT.DSU	DSU-110
ARMCortexA715CT.DSU.PPU_cluster	PPUv1
ARMCortexA715CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA715CT.DSU.PPU_core0	PPUv1
ARMCortexA715CT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexA715CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA715CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA715CT.DSU.shared_cache	PVCache
ARMCortexA715CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA715CT.DSU.shared_cache.upstream[Y] (where Y = 0-3)	PVBusSlave
ARMCortexA715CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA715CT.MMAP	PVBusLogger
ARMCortexA715CT.MMAP.mapper	PVBusMapper
ARMCortexA715CT.RAS	PVBusLogger
ARMCortexA715CT.RAS.mapper	PVBusMapper
ARMCortexA715CT.cpu0	ARM_Cortex-A715
ARMCortexA715CT.cpu0.UTLB	TLB
ARMCortexA715CT.cpu0.l1dcache	PVCache
ARMCortexA715CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA715CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA715CT.cpu0.l1licache	PVCache
ARMCortexA715CT.cpu0.l1licache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA715CT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA715CT.cpu0.l2cache	PVCache
ARMCortexA715CT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA715CT.cpu0.l2cache.upstream[U] (where U = 0-1)	PVBusSlave
ARMCortexA715CT.ext_bus	PVBusLogger
ARMCortexA715CT.ext_bus.mapper	PVBusMapper
ARMCortexA715CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

## Ports for ARMCortexA715CT

Port	Direction	Protocol	Description
acp_s	slave	PVBus	AXI ACP slave port.
AEND0MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND2MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).

Port	Direction	Protocol	Description
AEND3MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART2MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	slave	Signal	CHI defined pins.
broadcastcachemaint	slave	Signal	ACE defined pins.
broadcastouter	slave	Signal	ACE defined pins.
broadcastpersist	slave	Signal	CHI defined pins.
cfgend	slave	Signal	This signal is for EE bit initialisation.
cfgsdisable	slave	Signal	This signal disables write access to some secure Interrupt Controller registers.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcsn_pchannel	master	PChannel	Cluster PCSM signal
cluster_powerdown_out	master	Signal	DEPRECATED - An output from PPUs that informs thermal controller of the cluster power info
cluster_ppu_hw_stat	master	Value	Cluster PPU power state
clusterid	slave	Value	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set ManagerID64 of the core.
clusterpmuirq	master	Signal	DynamiQ pmu irq
CNTHPIRQ	master	Signal	Timer signals to SOC.
CNTHPSIRQ	master	Signal	Timer signals to SOC
CNTHVIRQ	master	Signal	Timer signals to SOC.
CNTHVSIQ	master	Signal	Timer signals to SOC
CNTPNSIRQ	master	Signal	Timer signals to SOC.
CNTPSIQ	master	Signal	Timer signals to SOC.
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.
CNTVIRQ	master	Signal	Timer signals to SOC.
commirq	master	Signal	Interrupt signal from debug communications channel.



Port	Direction	Protocol	Description
core_clk_in	slave	ClockSignal	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
core_pcsn_pchannel	master	PChannel	Core PCSM signals
core_powerdown_out	master	Signal	DEPRECATED - An output from PPU that informs thermal controller of the core power info
core_ppu_hw_stat	master	Value	Notify the power state change inside the PPU.
coreerrirq	master	Signal	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq	master	Signal	Fault indicator for a detected 1 or 2 bit ECC error
coreinstrrun	master	Signal	Core Running State
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
ctidbgirq	master	Signal	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	slave	Signal	External debug interface.
dbgnopwrdown	master	Signal	No power-down request.
dbgpwrupreq	master	Signal	Debug power up request.
defaultMP	slave	Signal	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main master interface ports.
dev_debug_s	slave	PVBus	External debug interface.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.
gicreset	master	Signal	An output from PPU that can be used to reset GICCLK domain components
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	slave	PVBus	External debug interface.
pmbirq	master	Signal	Interrupt signal from the statistical profiling unit.
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	master	Signal	PPU cluster interrupt.
ppu_cluster_isolate	master	Signal	PPU Cluster Domain isolation control.
ppu_cluster_wakerequest	slave	Signal	PPU cluster wake request signal.
ppu_core_irq	master	Signal	PPU core interrupt.
ppu_core_isolate	master	Signal	PPU Core Domain isolation control.
ppu_core_wakerequest	slave	Signal	PPU Core wake request signals.
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	master	PVBus	The core will generate bus requests on this port.

Port	Direction	Protocol	Description
pvbus_periph_m	master	PVBus	The core can generate peripheral bus request on this port.
rei	slave	Signal	Per core RAM Error Interrupt.
reset	slave	Signal	Reset signal to cluster.
rvbaraddr	slave	Value_64	Reset vector base address.
sci_m	master	SystemCoherencyInterface	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei	slave	Signal	Per core System Error physical pins.
spiden	slave	Signal	External debug interface.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq	master	Signal	Interrupt signal from the trace buffer unit.
utility_bus_s	slave	PVBus	Utility bus slave
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfiq	slave	Signal	Virtualised FIQ.
virq	slave	Signal	Virtualised IRQ.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Per core virtual System Error physical pins.

## Parameters for ARM Cortex A715CT

### cpuX.CFGEND

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### cpuX.CFGTE

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### cpuX.CRYPTODISABLE

Disable cryptographic features.

Type: `bool`

Default value: `false`

**cpuX.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**cpuX.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**cpuX.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**cpuX.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**cpuX.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**cpuX.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**cpuX.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**cpuX.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

### **`cpuX.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

### **`cpuX.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`cpuX.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`cpuX.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**cpuX.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**cpuX.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**cpuX.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**cpuX.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**cpuX.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**cpuX.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: 0x0

**cpuX.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: 0xf000000

**cpuX.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**cpuX.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**cpuX.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**cpuX.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**cpuX.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**AEND0\_DEFAULT**

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMPO input signal).

Type: `uint64_t`

Default value: `0x0`

**AEND1\_DEFAULT**

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

Type: `uint64_t`

Default value: `0x0`

**AEND2\_DEFAULT**

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

Type: `uint64_t`

Default value: `0x0`

**AEND3\_DEFAULT**

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART0\_DEFAULT**

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART1\_DEFAULT**

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART2\_DEFAULT**

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

Type: `uint64_t`

Default value: `0x0`



**ASTART3\_DEFAULT**

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

Type: `uint64_t`

Default value: `0x0`

**BROADCASTATOMIC**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`

Default value: `true`

**BROADCASTCACHEMAINT**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

Type: `bool`

Default value: `false`

**BROADCASTOUTER**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

Type: `bool`

Default value: `false`

**BROADCASTPERSIST**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: `bool`

Default value: `true`

**CLUSTER\_ID**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the ManagerID64 of the core.

Type: `uint32_t`

Default value: `0x0`

### **CMO\_broadcast\_when\_cache\_state\_modelling\_disabled**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. This value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required.

Type: `uint8_t`

Default value: `1`

### **CPUCFR**

Value of CPU Configuration Register.

Type: `uint64_t`

Default value: `0x0`

### **GICDISABLE**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`

Default value: `true`

### **NUM\_CORES**

Number of cores per cluster.

Type: `uint8_t`

Default value: `1`

### **bus\_type**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

Type: `uint8_t`

Default value: `0`

**core\_power\_on\_by\_default**

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPUs which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

Type: `bool`

Default value: `false`

**cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-maintenance\_latency**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-miss\_latency**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-prefetch\_enabled**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

Type: `bool`

Default value: `false`

**dcache-read\_access\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**dcache-size**

L1 D-Cache size in bytes.

Type: `uint32_t`

Default value: `0x8000`

**dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**dcache-state\_modelled**

Set whether D-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **`dcache-write_access_latency`**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`dcache-write_latency`**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`default_opmode`**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

Type: `uint8_t`

Default value: 4

### **`diagnostics`**

Enable DynamIQ diagnostic messages.

Type: `bool`

Default value: `false`

### **`enable_simulation_performance_optimizations`**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of `stage12_tlb_size` parameter to 1024).

Type: `bool`

Default value: `true`

**error\_record\_feature\_register**

RAS feature register values. An array of JSON objects. The JSON schema for the array is:

```
[{"ED":0x0, "IMPDEF_3_2":0x0, "UI":0x0, "FI":0x0, "UE":0x0, "CFI":0x0, "CEC":0x0, "RP":0x0, "DUI":0x0, "CEO":0x0, "CI":0x0, "TS":0x0, "INJ":0x0, "FRX":0x0, "UC":0x0, "UEU":0x0, "UER":0x0, "UEO":0x0, "DE":0x0, "CE":0x0, "Visibility":"Core"}, other_feature_register_values].
```

Where ED, UI, FI, CE and UE have valid values between 0x0 - 0x3. CFI and DUI have valid values 0x0, 0x2 and 0x3. CEC has valid values 0x0, 0x2 or 0x4. RP, CEO, INJ, FRX, UC, UEU, UER, UEO, DE has valid values 0x0 or 0x1. CI and TS has valid values of 0x0, 0x1 and 0x2. Visibility has valid values "Core" or "Cluster".

Type: string

Default value: "[{"ED":0x2, "IMPDEF\_3\_2":0x1, "UI":0x2, "FI":0x2, "UE":0x1, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x0, "INJ":0x1, "CI":0x2, "TS":0x0, "Visibility":"Cluster"}, {"ED":0x2, "UI":0x2, "FI":0x2, "UE":0x1, "CFI":0x2, "CEC":0x2, "RP":0x1, "DUI":0x0, "CEO":0x0, "INJ":0x1, "CI":0x0, "TS":0x0}]",

**ete.CLAIMTAGS**

Number of claim tags.

Type: uint8\_t

Default value: 4

**ete.MAX\_INST\_PER\_Q**

Maximum limit for the number of instructions implied by a Q element.

Type: uint16\_t

Default value: 0x1

**ete.NumberOfRSPairs**

Number of resource selector pairs.

Type: uint8\_t

Default value: 8

**ete.PIDR\_CM0D**

TRCPIDR CM0D value.

Type: uint8\_t

Default value: 0

**ete.PIDR\_REVAND**

TRCPIDR REVAND value.

Type: `uint8_t`

Default value: 0

**ete.PIDR\_REVISION**

TRCPIDR REVISION value.

Type: `uint8_t`

Default value: 0

**ete.Q\_CADENCE**

Number of instruction blocks traced between two Q elements.

Type: `uint16_t`

Default value: 0x1

**ete.RES0\_STATEFUL**

Whether **RES0** bits are stateful or **RAZ/WI**.

Type: `bool`

Default value: `false`

**ete.RETSTACK**

Return stack depth.

Type: `uint8_t`

Default value: 3

**ete.REVISION**

TRCIDR1 revision value.

Type: `uint8_t`

Default value: 1

**ete.SIM\_OVERFLOW\_GRANULARITY**

Number of instruction blocks in each granule, for simulated overflow.

Type: `uint32_t`

Default value: 0x64

**ete.SIM\_OVERFLOW\_PERCENTAGE**

Percentage of instruction blocks lost in each granule, for simulated overflow.

Type: `uint8_t`

Default value: 0

**ete.SOURCE\_ADDRESS**

Allow generation of source address elements.

Type: `bool`

Default value: `false`

**ete.TRACE\_OUTPUT**

File to which to write trace byte stream.

Type: `string`

Default value: N/A

**ete.TRCSRSTA\_FORCED\_EXCEP**

TRCSRSTA value for a forcibly traced exception.

Type: `bool`

Default value: `false`

**force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks**

Force MTE tag accesses to **RAZ/WI** and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

Type: `bool`

Default value: `false`

**force\_zero\_PSTATE\_PAN**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`



**force\_zero\_mpam\_partid\_and\_pmg**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

Type: `uint8_t`

Default value: 0

**has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

**has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

**has\_enhanced\_pan**

Implements Armv8.7 Enhanced PAN feature (FEAT\_PAN3) values of this parameter are:- 1, feature is implemented if ARMv8.7 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 2

**has\_ete**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (-plugin or -P).

Type: `bool`

Default value: `false`

**has\_large\_va**

Implement support for the extended 52-bit virtual addresses from ARMv8.2 (FEAT\_LVA) values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.2 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

**has\_peripheral\_port**

If true, an additional AXI peripheral port is configured.

Type: `bool`

Default value: `false`

**has\_statistical\_profiling**

Whether Statistical Based Profiling is implemented (FEAT\_SPE).

Type: `bool`

Default value: `false`

**icache-hit\_latency**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

**icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-size**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: `0x8000`

**icache-state\_modelled**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

**instruction\_tlb\_size**

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

Type: `uint32_t`

Default value: `0x0`

**invalidate\_code\_cache\_on\_icache\_cmo**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: `uint8_t`

Default value: 0

### **l3cache-has\_mpam**

L3 Cache has MPAM support.

Type: `bool`

Default value: `true`

### **l3cache-hit\_latency**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **l3cache-maintenance\_latency**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **l3cache-miss\_latency**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **l3cache-read\_access\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size `l3cache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **l3cache-read\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l3cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to

the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-size`**

L3 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

### **`l3cache-snoop_data_transfer_latency`**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-snoop_issue_latency`**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-ways`**

L3 Cache number of ways (sets are implicit from size).

Type: `uint16_t`

Default value: `16`

### **`l3cache-write_access_latency`**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size `l3cache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-write\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. l3cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**memory\_tagging\_support\_level**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

Type: uint8\_t

Default value: 3

**mpmm\_accumulator\_multiplier**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator.value of n means the accumulator will use (n \* accumulator value) to calculate the mpmm threshold (MPMM).is provided as a fast model workaround to handle cases where execution of quantums in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

Type: uint8\_t

Default value: 1

**num\_acp**

Number of ACP ports.

Type: uint8\_t

Default value: 0

**num\_nodes**

Number of transport nodes. Zero implies direct-connect configuration.

Type: uint8\_t

Default value: 1

**pmu-num\_counters**

Number of PMU counters implemented.

Type: uint8\_t

Default value: 6

### **pseudo\_fault\_generation\_feature\_register**

ARMv8.4 Standard Pseudo-fault generation feature register values. JSON schema for the parameter value is: [{"OF":false, "UC":false, "UEU":false, "UER":false, "UEO":false, "DE":false, "CE":0x0, "CI":false, "ER":false, "PN":false, "AV":false, "MV":false, "SYN":false, "R":false, "NA":false}, other\_pseudo-fault\_generating\_features\_register\_values]. Where OF, UC, UEU, UER, UEO, DE, CI, ER, PN, AV, MV, SYN, and R have valid false(NOT\_SUPPORTED) and true(FEATURE\_CONTROLLABLE), where CE can have 0(NOT\_SUPPORTED), 1(NONSPECIFIC\_CE\_SUPPORTED) and 3(TRANSIENT\_OR\_PERSISTENT\_CE\_SUPPORTED) and NA can have false(component fakes detection on next access) or true(component fakes detection spontaneously). Effective only when ERXFR's INJ field allows it or has\_ras\_fault\_injection is true.

Type: string

Default value: "[{"OF":true, "UC":true, "UEU":false, "UER":false, "UEO":false, "DE":0x1, "CE":0x1, "CI":true, "ER":false, "PN":true, "AV":false, "MV":true, "SYN":true, "R":true}, {"OF":false, "UC":true, "UEU":false, "UER":false, "UEO":false, "DE":0x1, "CE":0x1, "CI":false, "ER":false, "PN":false, "AV":false, "MV":false, "SYN":false, "R":true}]"

### **ptw\_latency**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

### **stage12\_tlb\_size**

Number of stage1+2 tlb entries. Valid values are 0 or >= 4.

Type: uint32\_t

Default value: 0x80

### **tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

### **tlbi\_stall\_enabled**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: bool

Default value: `false`

### **`treat_PAC_as_NOP`**

Non-architecture parameter. Treat Pointer Authentication as **NOP**. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are **NOP**. Brach, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

Type: `bool`

Default value: `false`

### **`walk_cache_latency`**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

## 3.66 ARMCortexA720AECT

Defined in `LISA/ARMCortexA720AECT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following parameters were added:

- `has_delayed_dbgreg`
- `has_delayed_sysreg`
- `num_acp`

### AE-specific features implemented

The only AE-specific features implemented in the model are Split-Lock and Lock-Step, with the following limitations:

- The implementation of Split-Lock and Lock-Step in the model is simply a halving of the number of cores available.
- Lock-step is implemented by a parameter that can be true or false for the entire execution only, rather than by a signal that can be enabled during execution.



As the model does not implement true Split-Lock functionality, the ports and parameters are contiguous. For example, for a 4-core CPU with Split-Lock enabled, `cpu0` and `cpu1` identify the available cores and associated ports, not `cpu0` and `cpu2`.

- Hybrid mode is not modeled in the DSU.

## Iris and MTI instances for ARMCortexA720AECT

This model has the following Iris instances:

Name	Instance type
ARMCortexA720AECT	Cluster_ARM_Cortex-A720AE
ARMCortexA720AECT.AMU	PVBusLogger
ARMCortexA720AECT.AMU.mapper	PVBusMapper
ARMCortexA720AECT.DAP	PVBusLogger
ARMCortexA720AECT.DAP.mapper	PVBusMapper
ARMCortexA720AECT.DSU	DSU-120
ARMCortexA720AECT.DSU.PPU_cluster	PPUv1
ARMCortexA720AECT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA720AECT.DSU.PPU_core0	PPUv1
ARMCortexA720AECT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexA720AECT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA720AECT.DSU.mpam_busslave	PVBusSlave
ARMCortexA720AECT.DSU.shared_cache	PVCache
ARMCortexA720AECT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA720AECT.DSU.shared_cache.upstream[Y] (where Y = 0-4)	PVBusSlave
ARMCortexA720AECT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA720AECT.MMAP	PVBusLogger
ARMCortexA720AECT.MMAP.mapper	PVBusMapper
ARMCortexA720AECT.RAS	PVBusLogger
ARMCortexA720AECT.RAS.mapper	PVBusMapper
ARMCortexA720AECT.cpu0	ARM_Cortex-A720AE
ARMCortexA720AECT.cpu0.UTLB	TLB
ARMCortexA720AECT.cpu0.debug_rom	debug_rom
ARMCortexA720AECT.cpu0.dtlb	TLB
ARMCortexA720AECT.cpu0.l1dcache	PVCache
ARMCortexA720AECT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA720AECT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA720AECT.cpu0.l1licache	PVCache
ARMCortexA720AECT.cpu0.l1licache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA720AECT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA720AECT.cpu0.l2cache	PVCache
ARMCortexA720AECT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster

Name	Instance type
ARMCortexA720AECT.cpu0.l2cache.upstream[U] (where U = 0-1)	PVBusSlave
ARMCortexA720AECT.ext_bus	PVBusLogger
ARMCortexA720AECT.ext_bus.mapper	PVBusMapper
ARMCortexA720AECT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA720AECT.global_debug_rom	debug_rom
ARMCortexA720AECT.secondary_debug_rom	debug_rom
ARMCortexA720AECT.sve	ScalableVectorExtension

This model has the following MTI trace components:

Name	Component type
ARMCortexA720AECT.AMU	PVBusLogger
ARMCortexA720AECT.AMU.mapper	PVBusMapper
ARMCortexA720AECT.DAP	PVBusLogger
ARMCortexA720AECT.DAP.mapper	PVBusMapper
ARMCortexA720AECT.DSU	DSU-120
ARMCortexA720AECT.DSU.PPU_cluster	PPUV1
ARMCortexA720AECT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA720AECT.DSU.PPU_core0	PPUV1
ARMCortexA720AECT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexA720AECT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA720AECT.DSU.mpam_busslave	PVBusSlave
ARMCortexA720AECT.DSU.shared_cache	PVCache
ARMCortexA720AECT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA720AECT.DSU.shared_cache.upstream[Y] (where Y = 0-4)	PVBusSlave
ARMCortexA720AECT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA720AECT.MMAP	PVBusLogger
ARMCortexA720AECT.MMAP.mapper	PVBusMapper
ARMCortexA720AECT.RAS	PVBusLogger
ARMCortexA720AECT.RAS.mapper	PVBusMapper
ARMCortexA720AECT.cpu0	ARM_Cortex-A720AE
ARMCortexA720AECT.cpu0.UTLB	TLB
ARMCortexA720AECT.cpu0.l1dcache	PVCache
ARMCortexA720AECT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA720AECT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA720AECT.cpu0.l1icache	PVCache
ARMCortexA720AECT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA720AECT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA720AECT.cpu0.l2cache	PVCache
ARMCortexA720AECT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA720AECT.cpu0.l2cache.upstream[U] (where U = 0-1)	PVBusSlave

Name	Component type
ARMCortexA720AECT.ext_bus	PVBusLogger
ARMCortexA720AECT.ext_bus.mapper	PVBusMapper
ARMCortexA720AECT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

## Ports for ARMCortexA720AECT

Port	Direction	Protocol	Description
acp_s	slave	PVBus	AXI ACP slave port.
AEND0MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND2MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND3MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART2MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	slave	Signal	CHI defined pins.
broadcastcachemaint	slave	Signal	ACE defined pins.
broadcastouter	slave	Signal	ACE defined pins.
broadcastpersist	slave	Signal	CHI defined pins.
cfgend	slave	Signal	This signal is for EE bit initialisation.
cfgsdisable	slave	Signal	This signal disables write access to some secure Interrupt Controller registers.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcsn_pchannel	master	PChannel	Cluster PCSM signal
cluster_powerdown_out	master	Signal	DEPRECATED - An output from PPU that informs thermal controller of the cluster power info
cluster_ppu_hw_stat	master	Value	Cluster PPU power state
clustercriticalirq	master	Signal	Cluster Critical Irq
clustererrirq	master	Signal	Cluster Error Irq
clusterfaultirq	master	Signal	Cluster Fault Irq

Port	Direction	Protocol	Description
clusterid	slave	Value	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set ManagerID64 of the core.
clusterpmuirq	master	Signal	DynamiQ pmu irq
CNTHPIRQ	master	Signal	Timer signals to SOC.
CNTHPSIRQ	master	Signal	Timer signals to SOC
CNTHVIRQ	master	Signal	Timer signals to SOC.
CNTHVSIQ	master	Signal	Timer signals to SOC
CNTPNSIRQ	master	Signal	Timer signals to SOC.
CNTPSIQ	master	Signal	Timer signals to SOC.
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.
CNTVIRQ	master	Signal	Timer signals to SOC.
commirq	master	Signal	Interrupt signal from debug communications channel.
core_clk_in	slave	ClockSignal	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
core_pcsn_pchannel	master	PChannel	Core PCSM signals
core_powerdown_out	master	Signal	DEPRECATED - An output from PPUs that informs thermal controller of the core power info
core_ppu_hw_stat	master	Value	Notify the power state change inside the PPU.
coreerrirq	master	Signal	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq	master	Signal	Fault indicator for a detected 1 or 2 bit ECC error
coreinstrrun	master	Signal	Core Running State
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
ctidbgirq	master	Signal	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	slave	Signal	External debug interface.
dbgnopwrdown	master	Signal	No power-down request.
dbgprupreq	master	Signal	Debug power up request.
defaultMP	slave	Signal	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main master interface ports.
dev_debug_s	slave	PVBus	External debug interface.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.

Port	Direction	Protocol	Description
gicreset	master	Signal	An output from PPUs that can be used to reset GICCLK domain components
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	slave	PVBus	External debug interface.
mpam_err_irq_ns	master	Signal	-
mpam_err_irq_s	master	Signal	MPAM Error signals
pmbirq	master	Signal	Interrupt signal from the statistical profiling unit.
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	master	Signal	PPU cluster interrupt.
ppu_cluster_isolate	master	Signal	PPU Cluster Domain isolation control.
ppu_cluster_wakerequest	slave	Signal	PPU cluster wake request signal.
ppu_core_irq	master	Signal	PPU core interrupt.
ppu_core_isolate	master	Signal	PPU Core Domain isolation control.
ppu_core_wakerequest	slave	Signal	PPU core wake request signal.
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	master	PVBus	The core will generate bus requests on this port.
pvbus_periph_m	master	PVBus	The core can generate peripheral bus request on this port.
rei	slave	Signal	Per core RAM Error Interrupt.
reset	slave	Signal	Reset signal to cluster.
rvbaraddr	slave	Value_64	Reset vector base address.
sci_m	master	SystemCoherencyInterface	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei	slave	Signal	Per core System Error physical pins.
spiden	slave	Signal	External debug interface.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq	master	Signal	Interrupt signal from the trace buffer unit.
utility_bus_s	slave	PVBus	Utility bus slave
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfiq	slave	Signal	Virtualised FIQ.
virq	slave	Signal	Virtualised IRQ.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Per core virtual System Error physical pins.

## Parameters for ARMCortexA720AECT

### **cpuX.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **cpuX.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **cpuX.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **cpuX.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **cpuX.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **cpuX.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**cpuX.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**cpuX.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**cpuX.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t



Default value: 0

**cpuX.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**cpuX.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**cpuX.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**cpuX.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**cpuX.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**cpuX.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**cpuX.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**cpuX.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: true

**cpuX.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**cpuX.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**cpuX.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**cpuX.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**cpuX.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**cpuX.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**cpuX.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**AEND0\_DEFAULT**

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

Type: `uint64_t`

Default value: `0x0`

**AEND1\_DEFAULT**

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

Type: `uint64_t`

Default value: `0x0`

**AEND2\_DEFAULT**

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

Type: `uint64_t`

Default value: `0x0`

**AEND3\_DEFAULT**

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART0\_DEFAULT**

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART1\_DEFAULT**

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART2\_DEFAULT**

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART3\_DEFAULT**

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

Type: `uint64_t`

Default value: `0x0`

**BROADCASTATOMIC**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`

Default value: `true`

**BROADCASTCACHEMAINT**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

Type: `bool`

Default value: `false`

**BROADCASTOUTER**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

Type: `bool`

Default value: `false`

**BROADCASTPERSIST**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: `bool`

Default value: `true`

**CLUSTER\_ID**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the ManagerID64 of the core.

Type: `uint32_t`

Default value: `0x0`

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. This value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required.

Type: `uint8_t`

Default value: `1`

**CPUCFR**

Value of CPU Configuration Register.

Type: `uint64_t`

Default value: `0x0`

**GICDISABLE**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`

Default value: `true`

**NUM\_CORES**

Number of cores per cluster.

Type: `uint8_t`

Default value: `1`

**bus\_type**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

Type: `uint8_t`

Default value: `0`

**cluster\_split\_lock\_config**

Default SPLIT/LOCKED config. The valid values are: 1 - Only LOCKED mode support, 4 - Only SPLIT mode support, 5 - SPLIT or MIXED mode support. Valid only when `enable_ae_features` is `true`.

Type: `uint8_t`

Default value: `1`

**core\_power\_on\_by\_default**

If `true`, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPUs which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

Type: `bool`

Default value: `false`

**cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-maintenance\_latency**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-miss\_latency**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-prefetch\_enabled**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

Type: `bool`

Default value: `false`

**dcache-read\_access\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-size**

L1 D-Cache size in bytes.

Type: `uint32_t`

Default value: `0x10000`

**dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-state\_modelled**

Set whether D-cache has stateful implementation.

Type: `bool`

Default value: `false`

**dcache-write\_access\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`



Default value: 0x0

### **default\_opmode**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

Type: `uint8_t`

Default value: 4

### **diagnostics**

Enable DynamIQ diagnostic messages.

Type: `bool`

Default value: `false`

### **enable\_lock\_step**

Whether the core is configured in Dual Core Lock Step mode (FEAT\_DCLS).

Type: `bool`

Default value: `false`

### **enable\_simulation\_performance\_optimizations**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of `stage12_tlb_size` parameter to 1024).

Type: `bool`

Default value: `true`

### **ete.CLAIMTAGS**

Number of claim tags.

Type: `uint8_t`

Default value: 4

### **ete.ETE\_REVISION**

ETE revision: 0=ETEv1.0, 1=ETEv1.1, 2=ETEv1.2, 3=ETEv1.3.

Type: `uint8_t`

Default value: 1

**ete.MAX\_INST\_PER\_Q**

Maximum limit for the number of instructions implied by a Q element.

Type: `uint16_t`

Default value: `0x1`

**ete.NumberOfRSPairs**

Number of resource selector pairs.

Type: `uint8_t`

Default value: `8`

**ete.PIDR\_CMOD**

TRCPIDR CMOD value.

Type: `uint8_t`

Default value: `0`

**ete.PIDR\_REVAND**

TRCPIDR REVAND value.

Type: `uint8_t`

Default value: `0`

**ete.PIDR\_REVISION**

TRCPIDR REVISION value.

Type: `uint8_t`

Default value: `0`

**ete.Q\_CADENCE**

Number of instruction blocks traced between two Q elements.

Type: `uint16_t`

Default value: `0x1`

**ete.RES0\_STATEFUL**

Whether **RES0** bits are stateful or **RAZ/WI**.

Type: `bool`

Default value: `false`

**ete.RETSTACK**

Return stack depth.

Type: `uint8_t`

Default value: 3

**ete.REVISION**

TRCIDR1 revision value.

Type: `uint8_t`

Default value: 0

**ete.SIM\_OVERFLOW\_GRANULARITY**

Number of instruction blocks in each granule, for simulated overflow.

Type: `uint32_t`

Default value: 0x64

**ete.SIM\_OVERFLOW\_PERCENTAGE**

Percentage of instruction blocks lost in each granule, for simulated overflow.

Type: `uint8_t`

Default value: 0

**ete.SOURCE\_ADDRESS**

Allow generation of source address elements.

Type: `bool`

Default value: `false`

**ete.TRACE\_OUTPUT**

File to which to write trace byte stream.

Type: `string`

Default value: `N/A`

**ete.TRCSRSTA\_FORCED\_EXCEP**

TRCSR.TA value for a forcibly traced exception.

Type: `bool`

Default value: `false`

**ete.TSMARK**

Whether timestamp markers are supported.

Type: `bool`

Default value: `true`

**force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks**

Force MTE tag accesses to **RAZ/WI** and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

Type: `bool`

Default value: `false`

**force\_zero\_PSTATE\_PAN**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

**force\_zero\_mpam\_partid\_and\_pmg**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAMO\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

Type: `uint8_t`

Default value: `0`

**has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

**has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

**has\_enhanced\_pan**

Implements Armv8.7 Enhanced PAN feature (FEAT\_PAN3) values of this parameter are:- 1, feature is implemented if ARMv8.7 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 2

**has\_ete**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (-plugin or -P).

Type: `bool`

Default value: `false`

**has\_impdef\_transient\_fault\_protection**

Support the Transient Fault Protection (TFP) flop parity errors through RAS registers (FEAT\_TFP).

Type: `bool`

Default value: `true`

**has\_large\_va**

Implement support for the extended 52-bit virtual addresses from ARMv8.2 (FEAT\_LVA) values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.2 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

**has\_peripheral\_port**

If true, an additional AXI peripheral port is configured.

Type: `bool`

Default value: `false`

**has\_statistical\_profiling**

Whether Statistical Based Profiling is implemented (FEAT\_SPE).

Type: `bool`

Default value: `false`

**icache-hit\_latency**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

**icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to

the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`icache-size`**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: `0x10000`

### **`icache-state_modelled`**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **`instruction_tlb_size`**

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

Type: `uint32_t`

Default value: `0x0`

### **`invalidate_code_cache_on_icache_cmo`**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: `uint8_t`

Default value: `0`

### **`l3cache-has_mpam`**

L3 Cache has MPAM support.

Type: `bool`

Default value: `true`

### **`l3cache-hit_latency`**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **l3cache-maintenance\_latency**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **l3cache-miss\_latency**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **l3cache-read\_access\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size `l3cache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **l3cache-read\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l3cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **l3cache-size**

L3 Cache size in bytes.

Type: `uint32_t`

Default value: 0x80000



**l3cache-snoop\_data\_transfer\_latency**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-snoop\_issue\_latency**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-ways**

L3 Cache number of ways (sets are implicit from size).

Type: `uint16_t`

Default value: `16`

**l3cache-write\_access\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size `l3cache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-write\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l3cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**memory\_tagging\_support\_level**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

Type: `uint8_t`

Default value: 3

**mpmm\_accumulator\_multiplier**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator.value of n means the accumulator will use (n \* accumulator value) to calculate the mpmm threshold (MPMM).is provided as a fast model workaround to handle cases where execution of quantum in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

Type: uint8\_t

Default value: 1

**num\_acp**

Number of ACP ports.

Type: uint8\_t

Default value: 0

**num\_nodes**

Number of transport nodes. Zero implies direct-connect configuration.

Type: uint8\_t

Default value: 1

**pmu-num\_counters**

Number of PMU counters implemented.

Type: uint8\_t

Default value: 6

**ptw\_latency**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

**reported\_patch\_level**

Purely cosmetic patch level value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: int8\_t

Default value: -1

**reported\_revision\_number**

Purely cosmetic revision number value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: `int8_t`

Default value: -1

**stage12\_tlb\_size**

Number of stage1+2 tlb entries. Valid values are 0 or  $\geq 4$ .

Type: `uint32_t`

Default value: 0x80

**tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

**tlbi\_stall\_enabled**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: `bool`

Default value: `false`

**treat\_PAC\_as\_NOP**

Non-architecture parameter. Treat Pointer Authentication as **NOP**. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are **NOP**. Branch, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

Type: `bool`

Default value: `false`

**walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

## 3.67 ARMCortexA720CT

Defined in `LISA/ARMCortexA720CT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p1	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following parameters were added:

- `has_delayed_dbgreg`
- `has_delayed_sysreg`

### About ARMCortexA720CT

The model supports the following features:

- L2 cache is supported at the per-core level only and there is no implementation of Core-Complex with shared L2 cache yet.
- Internal PPU support is present.
- A P-Channel for the cluster and for each core.
- `BROADCASTPERSIST` pin.
- Optional peripheral port.
- Memory-mapped register access to MPAM.
- Per-core clock.
- Utility bus.

Support for the following features is planned for a future release:

- DSU-120 system features are not fully implemented.
- Core-Complex.
- `BROADCASTCACHEMAINTPOU` pin
- `COREINSTRRET` and `COREINSTRRUN` signals.
- Each thread has its own L1Cache and L2Cache instead of a common cache that is shared by all threads of the core.
- Per-thread parameters are not supported but signals are implemented.

The following features will not be implemented:

- DynamIQ features that are irrelevant to the programmers' view simulation, for example:
  - Automatic CPU retention mode.

- Level-3 Cache RAM retention.
- 256-bit wide output transactions.
- ECC and parity schemes are hardware-specific so are not supported.
- Self-test features (MBIST).
- Latency configuration.
- Snoop filtering.
- Cache stashing capability.

This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (crypto.dll or crypto.so) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

### Iris and MTI instances for ARM Cortex A720CT

This model has the following Iris instances:

Name	Instance type
ARM Cortex A720CT	Cluster_ARM_Cortex-A720
ARM Cortex A720CT.AMU	PVBusLogger
ARM Cortex A720CT.AMU.mapper	PVBusMapper
ARM Cortex A720CT.DAP	PVBusLogger
ARM Cortex A720CT.DAP.mapper	PVBusMapper
ARM Cortex A720CT.DSU	DSU-120
ARM Cortex A720CT.DSU.PPU_cluster	PPUv1
ARM Cortex A720CT.DSU.PPU_cluster.busslave	PVBusSlave
ARM Cortex A720CT.DSU.PPU_core0	PPUv1
ARM Cortex A720CT.DSU.PPU_core0.busslave	PVBusSlave
ARM Cortex A720CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARM Cortex A720CT.DSU.mpam_busslave	PVBusSlave
ARM Cortex A720CT.DSU.shared_cache	PVCache
ARM Cortex A720CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARM Cortex A720CT.DSU.shared_cache.upstream[Y] (where Y = 0-4)	PVBusSlave
ARM Cortex A720CT.DSU.utility_slave[0]	PVBusSlave
ARM Cortex A720CT.MMAP	PVBusLogger
ARM Cortex A720CT.MMAP.mapper	PVBusMapper
ARM Cortex A720CT.RAS	PVBusLogger
ARM Cortex A720CT.RAS.mapper	PVBusMapper
ARM Cortex A720CT.cpu0	ARM_Cortex-A720
ARM Cortex A720CT.cpu0.UTLB	TLB
ARM Cortex A720CT.cpu0.debug_rom	debug_rom
ARM Cortex A720CT.cpu0.dtlb	TLB
ARM Cortex A720CT.cpu0.l1dcache	PVCache

Name	Instance type
ARMCortexA720CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA720CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA720CT.cpu0.l1licache	PVCache
ARMCortexA720CT.cpu0.l1licache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA720CT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA720CT.cpu0.l2cache	PVCache
ARMCortexA720CT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA720CT.cpu0.l2cache.upstream[U] (where U = 0-1)	PVBusSlave
ARMCortexA720CT.ext_bus	PVBusLogger
ARMCortexA720CT.ext_bus.mapper	PVBusMapper
ARMCortexA720CT.gic_cpuif_decoder_cluster	GLCv3CPUInterfaceDecoder
ARMCortexA720CT.global_debug_rom	debug_rom
ARMCortexA720CT.secondary_debug_rom	debug_rom
ARMCortexA720CT.sve	ScalableVectorExtension

This model has the following MTI trace components:

Name	Component type
ARMCortexA720CT.AMU	PVBusLogger
ARMCortexA720CT.AMU.mapper	PVBusMapper
ARMCortexA720CT.DAP	PVBusLogger
ARMCortexA720CT.DAP.mapper	PVBusMapper
ARMCortexA720CT.DSU	DSU-120
ARMCortexA720CT.DSU.PPU_cluster	PPUv1
ARMCortexA720CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA720CT.DSU.PPU_core0	PPUv1
ARMCortexA720CT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexA720CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA720CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA720CT.DSU.shared_cache	PVCache
ARMCortexA720CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA720CT.DSU.shared_cache.upstream[Y] (where Y = 0-4)	PVBusSlave
ARMCortexA720CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA720CT.MMAP	PVBusLogger
ARMCortexA720CT.MMAP.mapper	PVBusMapper
ARMCortexA720CT.RAS	PVBusLogger
ARMCortexA720CT.RAS.mapper	PVBusMapper
ARMCortexA720CT.cpu0	ARM_Cortex-A720
ARMCortexA720CT.cpu0.UTLB	TLB
ARMCortexA720CT.cpu0.l1dcache	PVCache
ARMCortexA720CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster

Name	Component type
ARMCortexA720CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA720CT.cpu0.l1icache	PVCache
ARMCortexA720CT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA720CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA720CT.cpu0.l2cache	PVCache
ARMCortexA720CT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA720CT.cpu0.l2cache.upstream[U] (where U = 0-1)	PVBusSlave
ARMCortexA720CT.ext_bus	PVBusLogger
ARMCortexA720CT.ext_bus.mapper	PVBusMapper
ARMCortexA720CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

### Ports for ARMCortexA720CT

Port	Direction	Protocol	Description
acp_s	slave	PVBus	AXI ACP slave port.
AEND0MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND2MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND3MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART2MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	slave	Signal	CHI defined pins.
broadcastcachemaint	slave	Signal	ACE defined pins.
broadcastouter	slave	Signal	ACE defined pins.
broadcastpersist	slave	Signal	CHI defined pins.
cfgend	slave	Signal	This signal is for EE bit initialisation.
cfgsdisable	slave	Signal	This signal disables write access to some secure Interrupt Controller registers.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcsn_pchannel	master	PChannel	Cluster PCSM signal

Port	Direction	Protocol	Description
cluster_powerdown_out	master	Signal	DEPRECATED - An output from PPUs that informs thermal controller of the cluster power info
cluster_ppu_hw_stat	master	Value	Cluster PPU power state
clusterid	slave	Value	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set ManagerID64 of the core.
clusterpmuirq	master	Signal	DynamiQ pmu irq
CNTHPIRQ	master	Signal	Timer signals to SOC.
CNTHPSIRQ	master	Signal	Timer signals to SOC
CNTHVIRQ	master	Signal	Timer signals to SOC.
CNTHVSIRQ	master	Signal	Timer signals to SOC
CNTPNSIRQ	master	Signal	Timer signals to SOC.
CNTPSIRQ	master	Signal	Timer signals to SOC.
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.
CNTVIRQ	master	Signal	Timer signals to SOC.
commirq	master	Signal	Interrupt signal from debug communications channel.
core_clk_in	slave	ClockSignal	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
core_pcsn_pchannel	master	PChannel	Core PCSM signals
core_powerdown_out	master	Signal	DEPRECATED - An output from PPUs that informs thermal controller of the core power info
core_ppu_hw_stat	master	Value	Notify the power state change inside the PPU.
coreerrirq	master	Signal	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq	master	Signal	Fault indicator for a detected 1 or 2 bit ECC error
coreinstrrun	master	Signal	Core Running State
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
ctidbgirq	master	Signal	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	slave	Signal	External debug interface.
dbgnopwrdown	master	Signal	No power-down request.
dbgprupreq	master	Signal	Debug power up request.
defaultMP	slave	Signal	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main master interface ports.
dev_debug_s	slave	PVBus	External debug interface.



Port	Direction	Protocol	Description
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.
gicreset	master	Signal	An output from PPU that can be used to reset GICCLK domain components
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	slave	PVBus	External debug interface.
pmbirq	master	Signal	Interrupt signal from the statistical profiling unit.
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	master	Signal	PPU cluster interrupt.
ppu_cluster_isolate	master	Signal	PPU Cluster Domain isolation control.
ppu_cluster_wakerequest	slave	Signal	PPU cluster wake request signal.
ppu_core_irq	master	Signal	PPU core interrupt.
ppu_core_isolate	master	Signal	PPU Core Domain isolation control.
ppu_core_wakerequest	slave	Signal	PPU core wake request signal.
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	master	PVBus	The core will generate bus requests on this port.
pvbus_periph_m	master	PVBus	The core can generate peripheral bus request on this port.
rei	slave	Signal	Per core RAM Error Interrupt.
reset	slave	Signal	Reset signal to cluster.
rvbaraddr	slave	Value_64	Reset vector base address.
sci_m	master	SystemCoherencyInterface	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei	slave	Signal	Per core System Error physical pins.
spiden	slave	Signal	External debug interface.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq	master	Signal	Interrupt signal from the trace buffer unit.
utility_bus_s	slave	PVBus	Utility bus slave
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfiq	slave	Signal	Virtualised FIQ.
virq	slave	Signal	Virtualised IRQ.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Per core virtual System Error physical pins.

## Parameters for ARMCortexA720CT

### **cpuX.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **cpuX.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **cpuX.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **cpuX.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **cpuX.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **cpuX.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**cpuX.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**cpuX.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**cpuX.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**cpuX.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**cpuX.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**cpuX.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**cpuX.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**cpuX.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**cpuX.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**cpuX.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**cpuX.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: true

**cpuX.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**cpuX.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**cpuX.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**cpuX.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**cpuX.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**cpuX.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**cpuX.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**AEND0\_DEFAULT**

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

Type: `uint64_t`

Default value: `0x0`

**AEND1\_DEFAULT**

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

Type: `uint64_t`

Default value: `0x0`

**AEND2\_DEFAULT**

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

Type: `uint64_t`

Default value: `0x0`

**AEND3\_DEFAULT**

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART0\_DEFAULT**

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART1\_DEFAULT**

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART2\_DEFAULT**

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART3\_DEFAULT**

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

Type: `uint64_t`

Default value: `0x0`

**BROADCASTATOMIC**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`

Default value: `true`

**BROADCASTCACHEMAINT**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

Type: `bool`

Default value: `false`



**BROADCASTOUTER**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

Type: `bool`

Default value: `false`

**BROADCASTPERSIST**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: `bool`

Default value: `true`

**CLUSTER\_ID**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the ManagerID64 of the core.

Type: `uint32_t`

Default value: `0x0`

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. This value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required.

Type: `uint8_t`

Default value: `1`

**CPUCFR**

Value of CPU Configuration Register.

Type: `uint64_t`

Default value: `0x0`

**GICDISABLE**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`

Default value: `true`

**NUM\_CORES**

Number of cores per cluster.

Type: `uint8_t`

Default value: 1

**bus\_type**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

Type: `uint8_t`

Default value: 0

**core\_power\_on\_by\_default**

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPUs which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

Type: `bool`

Default value: `false`

**cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: 0x1

**cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: 0x1

**dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-maintenance\_latency**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-miss\_latency**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-prefetch\_enabled**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

Type: `bool`

Default value: `false`

**dcache-read\_access\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`dcache-size`**

L1 D-Cache size in bytes.

Type: `uint32_t`

Default value: `0x8000`

### **`dcache-snoop_data_transfer_latency`**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`dcache-state_modelled`**

Set whether D-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **`dcache-write_access_latency`**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`dcache-write_latency`**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`default_opmode`**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

Type: `uint8_t`

Default value: 4

### **diagnostics**

Enable DynamlQ diagnostic messages.

Type: `bool`

Default value: `false`

### **enable\_simulation\_performance\_optimizations**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of `stage12_tlb_size` parameter to 1024).

Type: `bool`

Default value: `true`

### **ete.CLAIMTAGS**

Number of claim tags.

Type: `uint8_t`

Default value: 4

### **ete.ETE\_REVISION**

ETE revision: 0=ETEv1.0, 1=ETEv1.1, 2=ETEv1.2, 3=ETEv1.3.

Type: `uint8_t`

Default value: 1

### **ete.MAX\_INST\_PER\_Q**

Maximum limit for the number of instructions implied by a Q element.

Type: `uint16_t`

Default value: 0x1

### **ete.NumberOfRSPairs**

Number of resource selector pairs.

Type: `uint8_t`

Default value: 8

**ete.PIDR\_CM0D**

TRCPIDR CM0D value.

Type: `uint8_t`

Default value: 0

**ete.PIDR\_REVAND**

TRCPIDR REVAND value.

Type: `uint8_t`

Default value: 0

**ete.PIDR\_REVISION**

TRCPIDR REVISION value.

Type: `uint8_t`

Default value: 0

**ete.Q\_CADENCE**

Number of instruction blocks traced between two Q elements.

Type: `uint16_t`

Default value: 0x1

**ete.RES0\_STATEFUL**

Whether **RES0** bits are stateful or **RAZ/WI**.

Type: `bool`

Default value: `false`

**ete.RETSTACK**

Return stack depth.

Type: `uint8_t`

Default value: 3

**ete.REVISION**

TRCIDR1 revision value.

Type: `uint8_t`

Default value: 0

**ete.SIM\_OVERFLOW\_GRANULARITY**

Number of instruction blocks in each granule, for simulated overflow.

Type: `uint32_t`

Default value: `0x64`

**ete.SIM\_OVERFLOW\_PERCENTAGE**

Percentage of instruction blocks lost in each granule, for simulated overflow.

Type: `uint8_t`

Default value: `0`

**ete.SOURCE\_ADDRESS**

Allow generation of source address elements.

Type: `bool`

Default value: `false`

**ete.TRACE\_OUTPUT**

File to which to write trace byte stream.

Type: `string`

Default value: `N/A`

**ete.TRCRSRTA\_FORCED\_EXCEP**

TRCRSR.TA value for a forcibly traced exception.

Type: `bool`

Default value: `false`

**ete.TSMARK**

Whether timestamp markers are supported.

Type: `bool`

Default value: `true`

**force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks**

Force MTE tag accesses to **RAZ/WI** and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

Type: `bool`

Default value: `false`

### **`force_zero_PSTATE_PAN`**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

### **`force_zero_mpam_partid_and_pmg`**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAMO\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

Type: `uint8_t`

Default value: `0`

### **`has_delayed_dbgreg`**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

### **`has_delayed_sysreg`**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

### **`has_enhanced_pan`**

Implements Armv8.7 Enhanced PAN feature (FEAT\_PAN3) values of this parameter are:- 1, feature is implemented if ARMv8.7 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: `2`

### **`has_ete`**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (-plugin or -P).

Type: `bool`



Default value: `false`

### **has\_large\_va**

Implement support for the extended 52-bit virtual addresses from ARMv8.2 (FEAT\_LVA). values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.2 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **has\_peripheral\_port**

If true, an additional AXI peripheral port is configured.

Type: `bool`

Default value: `false`

### **has\_statistical\_profiling**

Whether Statistical Based Profiling is implemented (FEAT\_SPE).

Type: `bool`

Default value: `false`

### **icache-hit\_latency**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

Type: `bool`

Default value: `false`

**icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when icache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**icache-size**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: `0x8000`

**icache-state\_modelled**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

**instruction\_tlb\_size**

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

Type: `uint32_t`

Default value: 0x0

### **invalidate\_code\_cache\_on\_icache\_cmo**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: uint8\_t

Default value: 0

### **l3cache-has\_mpam**

L3 Cache has MPAM support.

Type: bool

Default value: true

### **l3cache-hit\_latency**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **l3cache-maintenance\_latency**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **l3cache-miss\_latency**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **l3cache-read\_access\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size l3cache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

### **l3cache-read\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l3cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **l3cache-size**

L3 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

### **l3cache-snoop\_data\_transfer\_latency**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **l3cache-snoop\_issue\_latency**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **l3cache-ways**

L3 Cache number of ways (sets are implicit from size).

Type: `uint16_t`

Default value: `16`

### **l3cache-write\_access\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size `l3cache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used

instead of per-byte even if l3cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **l3cache-write\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. l3cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **memory\_tagging\_support\_level**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

Type: uint8\_t

Default value: 3

### **mpmm\_accumulator\_multiplier**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator.value of n means the accumulator will use (n \* accumulator value) to calculate the mpmm threshold (MPMM).is provided as a fast model workaround to handle cases where execution of quantum in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

Type: uint8\_t

Default value: 1

### **num\_acp**

Number of ACP ports.

Type: uint8\_t

Default value: 0

### **num\_nodes**

Number of transport nodes. Zero implies direct-connect configuration.

Type: uint8\_t

Default value: 1

**pmu-num\_counters**

Number of PMU counters implemented.

Type: `uint8_t`

Default value: 6

**ptw\_latency**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

**stage12\_tlb\_size**

Number of stage1+2 tlb entries. Valid values are 0 or >= 4.

Type: `uint32_t`

Default value: 0x80

**tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

**tlbi\_stall\_enabled**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: `bool`

Default value: `false`

**treat\_PAC\_as\_NOP**

Non-architecture parameter. Treat Pointer Authentication as **NOP**. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are **NOP**. Brach, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

Type: `bool`

Default value: `false`

**walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

3.68 ARMCortexA725CT

Defined in LISA/ARMCortexA725CT.lisa.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

Changes in 11.31.15

The following parameters were added:

- has\_delayed\_dbgreg
- has\_delayed\_sysreg
- num\_acp

About ARMCortexA725CT

A DSU-120 DynamIQ cluster containing a configurable number of Cortex-A725 cores.

The number of cores in the cluster is configurable using the following parameter:

**NUM\_CORES**

Possible values are 1-14

The following DSU/CPU features will not be implemented:

- DynamIQ features that are irrelevant to the programmers' view simulation.
- 256-bit wide output transactions.
- ECC and parity schemes are hardware-specific so are not supported.
- Self-test features (MBIST).
- Snoop filtering.
- Cache stashing capability.
- Embedded Logic Analyzer (ELA).

This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (crypto.dll or crypto.so) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

## Iris and MTI instances for ARMCortexA725CT

This model has the following Iris instances:

Name	Instance type
ARMCortexA725CT	Cluster_ARM_Cortex-A725
ARMCortexA725CT.AMU	PVBusLogger
ARMCortexA725CT.AMU.mapper	PVBusMapper
ARMCortexA725CT.DAP	PVBusLogger
ARMCortexA725CT.DAP.mapper	PVBusMapper
ARMCortexA725CT.DSU	DSU-120
ARMCortexA725CT.DSU.PPU_cluster	PPUv1
ARMCortexA725CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA725CT.DSU.PPU_core0	PPUv1
ARMCortexA725CT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexA725CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA725CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA725CT.DSU.shared_cache	PVCache
ARMCortexA725CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA725CT.DSU.shared_cache.upstream[Y] (where Y = 0-4)	PVBusSlave
ARMCortexA725CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA725CT.MMAP	PVBusLogger
ARMCortexA725CT.MMAP.mapper	PVBusMapper
ARMCortexA725CT.RAS	PVBusLogger
ARMCortexA725CT.RAS.mapper	PVBusMapper
ARMCortexA725CT.cpu0	ARM_Cortex-A725
ARMCortexA725CT.cpu0.UTLB	TLB
ARMCortexA725CT.cpu0.debug_rom	debug_rom
ARMCortexA725CT.cpu0.dtlb	TLB
ARMCortexA725CT.cpu0.l1dcache	PVCache
ARMCortexA725CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA725CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA725CT.cpu0.l1icache	PVCache
ARMCortexA725CT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA725CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA725CT.cpu0.l2cache	PVCache
ARMCortexA725CT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA725CT.cpu0.l2cache.upstream[U] (where U = 0-1)	PVBusSlave
ARMCortexA725CT.ext_bus	PVBusLogger



Name	Instance type
ARMCortexA725CT.ext_bus.mapper	PVBusMapper
ARMCortexA725CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA725CT.global_debug_rom	debug_rom
ARMCortexA725CT.secondary_debug_rom	debug_rom
ARMCortexA725CT.sve	ScalableVectorExtension

This model has the following MTI trace components:

Name	Component type
ARMCortexA725CT.AMU	PVBusLogger
ARMCortexA725CT.AMU.mapper	PVBusMapper
ARMCortexA725CT.DAP	PVBusLogger
ARMCortexA725CT.DAP.mapper	PVBusMapper
ARMCortexA725CT.DSU	DSU-120
ARMCortexA725CT.DSU.PPU_cluster	PPUv1
ARMCortexA725CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA725CT.DSU.PPU_core0	PPUv1
ARMCortexA725CT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexA725CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA725CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA725CT.DSU.shared_cache	PVCache
ARMCortexA725CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA725CT.DSU.shared_cache.upstream[Y] (where Y = 0-4)	PVBusSlave
ARMCortexA725CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA725CT.MMAP	PVBusLogger
ARMCortexA725CT.MMAP.mapper	PVBusMapper
ARMCortexA725CT.RAS	PVBusLogger
ARMCortexA725CT.RAS.mapper	PVBusMapper
ARMCortexA725CT.cpu0	ARM_Cortex-A725
ARMCortexA725CT.cpu0.UTLB	TLB
ARMCortexA725CT.cpu0.l1dcache	PVCache
ARMCortexA725CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA725CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA725CT.cpu0.l1icache	PVCache
ARMCortexA725CT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA725CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA725CT.cpu0.l2cache	PVCache
ARMCortexA725CT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA725CT.cpu0.l2cache.upstream[U] (where U = 0-1)	PVBusSlave
ARMCortexA725CT.ext_bus	PVBusLogger
ARMCortexA725CT.ext_bus.mapper	PVBusMapper

Name	Component type
ARMCortexA725CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

## Ports for ARMCortexA725CT

Port	Direction	Protocol	Description
acp_s	slave	PVBus	AXI ACP slave port.
AEND0MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND2MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND3MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART2MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	slave	Signal	CHI defined pins.
broadcastcachemaint	slave	Signal	ACE defined pins.
broadcastouter	slave	Signal	ACE defined pins.
broadcastpersist	slave	Signal	CHI defined pins.
cfgend	slave	Signal	This signal is for EE bit initialisation.
cfgsdisable	slave	Signal	This signal disables write access to some secure Interrupt Controller registers.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcsn_pchannel	master	PChannel	Cluster PCSM signal
cluster_powerdown_out	master	Signal	DEPRECATED - An output from PPUs that informs thermal controller of the cluster power info
cluster_ppu_hw_stat	master	Value	Cluster PPU power state
clusterid	slave	Value	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set ManagerID64 of the core.

Port	Direction	Protocol	Description
clusterpmuirq	master	Signal	DynamiQ pmu irq
CNTHPIRQ	master	Signal	Timer signals to SOC.
CNTHPSIRQ	master	Signal	Timer signals to SOC
CNTHVIRQ	master	Signal	Timer signals to SOC.
CNTHVSIRQ	master	Signal	Timer signals to SOC
CNTPNSIRQ	master	Signal	Timer signals to SOC.
CNTPSIRQ	master	Signal	Timer signals to SOC.
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.
CNTVIRQ	master	Signal	Timer signals to SOC.
commirq	master	Signal	Interrupt signal from debug communications channel.
core_clk_in	slave	ClockSignal	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
core_pcsmpchannel	master	PChannel	Core PCSM signals
core_powerdown_out	master	Signal	DEPRECATED - An output from PPUs that informs thermal controller of the core power info
core_ppu_hw_stat	master	Value	Notify the power state change inside the PPU.
coreerrirq	master	Signal	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq	master	Signal	Fault indicator for a detected 1 or 2 bit ECC error
coreinstrrun	master	Signal	Core Running State
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
ctidbgirq	master	Signal	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	slave	Signal	External debug interface.
dbgnopwrdown	master	Signal	No power-down request.
dbgprupreq	master	Signal	Debug power up request.
defaultMP	slave	Signal	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main master interface ports.
dev_debug_s	slave	PVBus	External debug interface.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.
gicreset	master	Signal	An output from PPUs that can be used to reset GICCLK domain components
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	slave	PVBus	External debug interface.
pmbirq	master	Signal	Interrupt signal from the statistical profiling unit.
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.

Port	Direction	Protocol	Description
ppu_cluster_irq	master	Signal	PPU cluster interrupt.
ppu_cluster_isolate	master	Signal	PPU Cluster Domain isolation control.
ppu_cluster_wakerequest	slave	Signal	PPU cluster wake request signal.
ppu_core_irq	master	Signal	PPU core interrupt.
ppu_core_isolate	master	Signal	PPU Core Domain isolation control.
ppu_core_wakerequest	slave	Signal	PPU core wake request signal.
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	master	PVBus	The core will generate bus requests on this port.
pvbus_periph_m	master	PVBus	The core can generate peripheral bus request on this port.
rei	slave	Signal	Per core RAM Error Interrupt.
reset	slave	Signal	Reset signal to cluster.
rvbaraddr	slave	Value_64	Reset vector base address.
sci_m	master	SystemCoherencyInterface	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei	slave	Signal	Per core System Error physical pins.
spiden	slave	Signal	External debug interface.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq	master	Signal	Interrupt signal from the trace buffer unit.
utility_bus_s	slave	PVBus	Utility bus slave
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfiq	slave	Signal	Virtualised FIQ.
virq	slave	Signal	Virtualised IRQ.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Per core virtual System Error physical pins.

## Parameters for ARM Cortex A725CT

### cpuX.CFGEND

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: false

### cpuX.CFGTE

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **`cpuX.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`cpuX.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`cpuX.force-fpsid`**

Override the FPSID value.

Type: `bool`

Default value: `true`

### **`cpuX.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**cpuX.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x100000

**cpuX.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **cpuX.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **cpuX.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **cpuX.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

### **cpuX.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

### **cpuX.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

### **cpuX.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**cpuX.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**cpuX.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**cpuX.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**cpuX.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**cpuX.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**cpuX.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**cpuX.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`



**cpuX.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**cpuX.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**cpuX.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**cpuX.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**cpuX.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**cpuX.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**AEND0\_DEFAULT**

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

Type: `uint64_t`

Default value: `0x0`

**AEND1\_DEFAULT**

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

Type: `uint64_t`

Default value: `0x0`

**AEND2\_DEFAULT**

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

Type: `uint64_t`

Default value: `0x0`

**AEND3\_DEFAULT**

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART0\_DEFAULT**

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART1\_DEFAULT**

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART2\_DEFAULT**

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART3\_DEFAULT**

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

Type: `uint64_t`

Default value: `0x0`

**BROADCASTATOMIC**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`

Default value: `true`

**BROADCASTCACHEMAINT**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

Type: `bool`

Default value: `false`

**BROADCASTOUTER**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

Type: `bool`

Default value: `false`

**BROADCASTPERSIST**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: `bool`

Default value: `true`

**CLUSTER\_ID**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the ManagerID64 of the core.

Type: `uint32_t`

Default value: `0x0`

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. This value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain events even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required.

Type: `uint8_t`

Default value: `1`

**CPUCFR**

Value of CPU Configuration Register.

Type: `uint64_t`

Default value: `0x0`

**GICDISABLE**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`

Default value: `true`

**NUM\_CORES**

Number of cores per cluster.

Type: `uint8_t`

Default value: `1`

**bus\_type**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

Type: `uint8_t`

Default value: 0

**core\_power\_on\_by\_default**

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPUs which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

Type: `bool`

Default value: `false`

**cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: 0x1

**cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: 0x1

**dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**dcache-maintenance\_latency**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**dcache-miss\_latency**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**dcache-prefetch\_enabled**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

Type: `bool`

Default value: `false`

**dcache-read\_access\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**dcache-size**

L1 D-Cache size in bytes.

Type: `uint32_t`

Default value: `0x10000`

**dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: 0x0

### **dcache-state\_modelled**

Set whether D-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **dcache-write\_access\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **default\_opmode**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

Type: `uint8_t`

Default value: 4

### **diagnostics**

Enable DynamIQ diagnostic messages.

Type: `bool`

Default value: `false`

### **enable\_simulation\_performance\_optimizations**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of `stage12_tlb_size` parameter to 1024).

Type: `bool`

Default value: `true`

### **ete.CLAIMTAGS**

Number of claim tags.

Type: `uint8_t`

Default value: 4

### **ete.ETE\_REVISION**

ETE revision: 0=ETEv1.0, 1=ETEv1.1, 2=ETEv1.2, 3=ETEv1.3.

Type: `uint8_t`

Default value: 1

### **ete.MAX\_INST\_PER\_Q**

Maximum limit for the number of instructions implied by a Q element.

Type: `uint16_t`

Default value: `0x1`

### **ete.NumberOfRSPairs**

Number of resource selector pairs.

Type: `uint8_t`

Default value: 8

### **ete.PIDR\_CM0D**

TRCPIDR CM0D value.

Type: `uint8_t`

Default value: 0

### **ete.PIDR\_REVAND**

TRCPIDR REVAND value.

Type: `uint8_t`

Default value: 0

### **ete.PIDR\_REVISION**

TRCPIDR REVISION value.



Type: `uint8_t`

Default value: 0

### **`ete.Q_CADENCE`**

Number of instruction blocks traced between two Q elements.

Type: `uint16_t`

Default value: 0x1

### **`ete.RES0_STATEFUL`**

Whether **RES0** bits are stateful or **RAZ/WI**.

Type: `bool`

Default value: `false`

### **`ete.RETSTACK`**

Return stack depth.

Type: `uint8_t`

Default value: 1

### **`ete.REVISION`**

TRCIDR1 revision value.

Type: `uint8_t`

Default value: 0

### **`ete.SIM_OVERFLOW_GRANULARITY`**

Number of instruction blocks in each granule, for simulated overflow.

Type: `uint32_t`

Default value: 0x64

### **`ete.SIM_OVERFLOW_PERCENTAGE`**

Percentage of instruction blocks lost in each granule, for simulated overflow.

Type: `uint8_t`

Default value: 0

### **`ete.SOURCE_ADDRESS`**

Allow generation of source address elements.

Type: `bool`

Default value: `false`

### **`ete.TRACE_OUTPUT`**

File to which to write trace byte stream.

Type: `string`

Default value: `N/A`

### **`ete.TRCSRSTA_FORCED_EXCEP`**

TRCSRSTA value for a forcibly traced exception.

Type: `bool`

Default value: `false`

### **`ete.TSMARK`**

Whether timestamp markers are supported.

Type: `bool`

Default value: `true`

### **`force_mte_tag_access_razwi_and_ignore_tag_checks`**

Force MTE tag accesses to **RAZ/WI** and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

Type: `bool`

Default value: `false`

### **`force_zero_PSTATE_PAN`**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

### **`force_zero_mpam_partid_and_pmg`**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAMO\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

Type: `uint8_t`

Default value: 0

### **has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

### **has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

### **has\_enhanced\_pan**

Implements Armv8.7 Enhanced PAN feature (FEAT\_PAN3) values of this parameter are:- 1, feature is implemented if ARMv8.7 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 2

### **has\_ete**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (-plugin or -P).

Type: `bool`

Default value: `false`

### **has\_large\_va**

Implement support for the extended 52-bit virtual addresses from ARMv8.2 (FEAT\_LVA). values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.2 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **has\_peripheral\_port**

If true, an additional AXI peripheral port is configured.

Type: `bool`

Default value: `false`

### **has\_statistical\_profiling**

Whether Statistical Based Profiling is implemented (FEAT\_SPE).

Type: `bool`

Default value: `false`

### **icache-hit\_latency**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

### **icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **icache-size**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: `0x10000`

### **icache-state\_modelled**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **instruction\_tlb\_size**

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

Type: `uint32_t`

Default value: `0x0`

### **invalidate\_code\_cache\_on\_icache\_cmo**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: `uint8_t`

Default value: `0`

### **l3cache-has\_mpam**

L3 Cache has MPAM support.

Type: `bool`

Default value: `true`

### **`l3cache-hit_latency`**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-maintenance_latency`**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-miss_latency`**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-read_access_latency`**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size `l3cache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-read_latency`**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l3cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-size**

L3 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

**l3cache-snoop\_data\_transfer\_latency**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-snoop\_issue\_latency**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-ways**

L3 Cache number of ways (sets are implicit from size).

Type: `uint16_t`

Default value: `16`

**l3cache-write\_access\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size `l3cache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-write\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l3cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**memory\_tagging\_support\_level**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

Type: `uint8_t`

Default value: 3

**mpmm\_accumulator\_multiplier**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator.value of n means the accumulator will use (n \* accumulator value) to calculate the mpmm threshold (MPMM).is provided as a fast model workaround to handle cases where execution of quantum in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

Type: `uint8_t`

Default value: 1

**num\_acp**

Number of ACP ports.

Type: `uint8_t`

Default value: 0

**pmu-num\_counters**

Number of PMU counters implemented.

Type: `uint8_t`

Default value: 6

**ptw\_latency**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

**stage12\_tlb\_size**

Number of stage1+2 tlb entries. Valid values are 0 or >= 4.

Type: `uint32_t`

Default value: 0x80



**tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

**tlbi\_stall\_enabled**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: `bool`

Default value: `false`

**treat\_PAC\_as\_NOP**

Non-architecture parameter. Treat Pointer Authentication as **NOP**. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are **NOP**. Branch, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

Type: `bool`

Default value: `false`

**walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

## 3.69 ARMCortexA725CT\_CortexX925CT

Defined in `LISA/ARMCortexA725CT_CortexX925CT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
CortexA725 r0p0	Full support
CortexX925 r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following parameters were added:

- `num_acp`
- `subcluster0.has_delayed_dbgreg`
- `subcluster0.has_delayed_sysreg`
- `subcluster1.has_delayed_dbgreg`
- `subcluster1.has_delayed_sysreg`

## About ARMCortexA720CT\_CortexX925CT

The number of cores in each subcluster is configurable using the following parameters:

### **subcluster0.NUM\_CORES**

Possible values are 1-13 (ARMCortexA720CT).

### **subcluster1.NUM\_CORES**

Possible values are 1-13 (ARMCortexX925CT).

The total number of cores in the cluster cannot exceed 14.

Port arrays are expanded to the width needed for the maximum number of cores in each subcluster. Use the following port array indexes:

- `<port_name>[0-12]` for cores in `subcluster0`.
- `<port_name>[13-25]` for cores in `subcluster1`.



All instances in the Master cross trigger matrix port array, `cti[26]` must be connected, regardless of the `num_cores` value used.

Core-specific parameters have the following prefixes:

- `subcluster0.cpu0` to `subcluster0.cpu12` identify cores in `subcluster0`.
- `subcluster1.cpu0` to `subcluster1.cpu12` identify cores in `subcluster1`.

For information about the cores in this model, see:

- [ARMCortexA720CT](#).
- [ARMCortexX925CT](#).

## Iris and MTI instances for ARMCortexA725CT\_CortexX925CT

This model has the following Iris instances:

Name	Instance type
ARMCortexA725CT_CortexX925CT	Cluster_ARM_CortexA725_CortexX925_Heterogeneous
ARMCortexA725CT_CortexX925CT.AMU	PVBusLogger
ARMCortexA725CT_CortexX925CT.AMU.mapper	PVBusMapper
ARMCortexA725CT_CortexX925CT.DAP	PVBusLogger

Name	Instance type
ARMCortexA725CT_CortexX925CT.DAP.mapper	PVBusMapper
ARMCortexA725CT_CortexX925CT.DSU	DSU-120
ARMCortexA725CT_CortexX925CT.DSU.PPU_cluster	PPUv1
ARMCortexA725CT_CortexX925CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA725CT_CortexX925CT.DSU.PPU_coreZ (where Z = 0-1)	PPUv1
ARMCortexA725CT_CortexX925CT.DSU.PPU_coreZ.busslave (where Z = 0-1)	PVBusSlave
ARMCortexA725CT_CortexX925CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA725CT_CortexX925CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA725CT_CortexX925CT.DSU.shared_cache	PVCache
ARMCortexA725CT_CortexX925CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA725CT_CortexX925CT.DSU.shared_cache.upstream[Z] (where Z = 0-6)	PVBusSlave
ARMCortexA725CT_CortexX925CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA725CT_CortexX925CT.MMAP	PVBusLogger
ARMCortexA725CT_CortexX925CT.MMAP.mapper	PVBusMapper
ARMCortexA725CT_CortexX925CT.RAS	PVBusLogger
ARMCortexA725CT_CortexX925CT.RAS.mapper	PVBusMapper
ARMCortexA725CT_CortexX925CT.cpuZ.debug_rom (where Z = 0-1)	debug_rom
ARMCortexA725CT_CortexX925CT.ext_bus	PVBusLogger
ARMCortexA725CT_CortexX925CT.ext_bus.mapper	PVBusMapper
ARMCortexA725CT_CortexX925CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA725CT_CortexX925CT.global_debug_rom	debug_rom
ARMCortexA725CT_CortexX925CT.secondary_debug_rom	debug_rom
ARMCortexA725CT_CortexX925CT.subcluster0	Subcluster_ARM_Cortex-A725
ARMCortexA725CT_CortexX925CT.subcluster0.cpu0	ARM_Cortex-A725
ARMCortexA725CT_CortexX925CT.subclusterZ.cpu0.UTLB (where Z = 0-1)	TLB
ARMCortexA725CT_CortexX925CT.subclusterZ.cpuU.dtlb (where Z = 0-1; U = 0-1)	TLB
ARMCortexA725CT_CortexX925CT.subclusterZ.cpu0.l1dcache (where Z = 0-1)	PVCache
ARMCortexA725CT_CortexX925CT.subclusterZ.cpu0.l1dcache.downstream[0].pvbusmaster (where Z = 0-1)	PVBusMaster
ARMCortexA725CT_CortexX925CT.subclusterZ.cpu0.l1dcache.upstream[0] (where Z = 0-1)	PVBusSlave
ARMCortexA725CT_CortexX925CT.subclusterZ.cpu0.l1licache (where Z = 0-1)	PVCache
ARMCortexA725CT_CortexX925CT.subclusterZ.cpu0.l1licache.downstream[0].pvbusmaster (where Z = 0-1)	PVBusMaster
ARMCortexA725CT_CortexX925CT.subclusterZ.cpu0.l1licache.upstream[0] (where Z = 0-1)	PVBusSlave

Name	Instance type
ARMCortexA725CT_CortexX925CT.subclusterZ.cpu0.l2cache (where Z = 0-1)	PVCache
ARMCortexA725CT_CortexX925CT.subclusterZ.cpu0.l2cache.downstream[0].pvbusmaster (where Z = 0-1)	PVBusMaster
ARMCortexA725CT_CortexX925CT.subclusterZ.cpu0.l2cache.upstream[W] (where Z = 0-1; W = 0-1)	PVBusSlave
ARMCortexA725CT_CortexX925CT.subclusterZ.sve (where Z = 0-1)	ScalableVectorExtension
ARMCortexA725CT_CortexX925CT.subcluster1	Subcluster_ARM_Cortex-X925
ARMCortexA725CT_CortexX925CT.subcluster1.cpu0	ARM_Cortex-X925

This model has the following MTI trace components:

Name	Component type
ARMCortexA725CT_CortexX925CT.AMU	PVBusLogger
ARMCortexA725CT_CortexX925CT.AMU.mapper	PVBusMapper
ARMCortexA725CT_CortexX925CT.DAP	PVBusLogger
ARMCortexA725CT_CortexX925CT.DAP.mapper	PVBusMapper
ARMCortexA725CT_CortexX925CT.DSU	DSU-120
ARMCortexA725CT_CortexX925CT.DSU.PPU_cluster	PPUv1
ARMCortexA725CT_CortexX925CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexA725CT_CortexX925CT.DSU.PPU_coreZ (where Z = 0-1)	PPUv1
ARMCortexA725CT_CortexX925CT.DSU.PPU_coreZ.busslave (where Z = 0-1)	PVBusSlave
ARMCortexA725CT_CortexX925CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA725CT_CortexX925CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA725CT_CortexX925CT.DSU.shared_cache	PVCache
ARMCortexA725CT_CortexX925CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA725CT_CortexX925CT.DSU.shared_cache.upstream[Z] (where Z = 0-6)	PVBusSlave
ARMCortexA725CT_CortexX925CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexA725CT_CortexX925CT.MMAP	PVBusLogger
ARMCortexA725CT_CortexX925CT.MMAP.mapper	PVBusMapper
ARMCortexA725CT_CortexX925CT.RAS	PVBusLogger
ARMCortexA725CT_CortexX925CT.RAS.mapper	PVBusMapper
ARMCortexA725CT_CortexX925CT.ext_bus	PVBusLogger
ARMCortexA725CT_CortexX925CT.ext_bus.mapper	PVBusMapper
ARMCortexA725CT_CortexX925CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA725CT_CortexX925CT.subcluster0.cpu0	ARM_Cortex-A725
ARMCortexA725CT_CortexX925CT.subclusterZ.cpu0.UTLB (where Z = 0-1)	TLB
ARMCortexA725CT_CortexX925CT.subclusterZ.cpu0.l1dcache (where Z = 0-1)	PVCache
ARMCortexA725CT_CortexX925CT.subclusterZ.cpu0.l1dcache.downstream[0].pvbusmaster (where Z = 0-1)	PVBusMaster
ARMCortexA725CT_CortexX925CT.subclusterZ.cpu0.l1dcache.upstream[0] (where Z = 0-1)	PVBusSlave

Name	Component type
ARMCortexA725CT_CortexX925CT.subclusterZ.cpu0.l1icache (where Z = 0-1)	PVCache
ARMCortexA725CT_CortexX925CT.subclusterZ.cpu0.l1icache.downstream[0].pvbusmaster (where Z = 0-1)	PVBusMaster
ARMCortexA725CT_CortexX925CT.subclusterZ.cpu0.l1icache.upstream[0] (where Z = 0-1)	PVBusSlave
ARMCortexA725CT_CortexX925CT.subclusterZ.cpu0.l2cache (where Z = 0-1)	PVCache
ARMCortexA725CT_CortexX925CT.subclusterZ.cpu0.l2cache.downstream[0].pvbusmaster (where Z = 0-1)	PVBusMaster
ARMCortexA725CT_CortexX925CT.subclusterZ.cpu0.l2cache.upstream[W] (where Z = 0-1; W = 0-1)	PVBusSlave
ARMCortexA725CT_CortexX925CT.subcluster1.cpu0	ARM_Cortex-X925

### Ports for ARMCortexA725CT\_CortexX925CT

Port	Direction	Protocol	Description
acp_s	slave	PVBus	AXI ACP slave port.
AEND0MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND2MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND3MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART2MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	slave	Signal	CHI defined pins.
broadcastcachemaint	slave	Signal	ACE defined pins.
broadcastouter	slave	Signal	ACE defined pins.
broadcastpersist	slave	Signal	CHI defined pins.
cfgend	slave	Signal	This signal is for EE bit initialisation.
cfgsdisable	slave	Signal	This signal disables write access to some secure Interrupt Controller registers.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcs_m_pchannel	master	PChannel	Cluster PCSM signal

Port	Direction	Protocol	Description
cluster_powerdown_out	master	Signal	DEPRECATED - An output from PPUs that informs thermal controller of the cluster power info
cluster_ppu_hw_stat	master	Value	Cluster PPU power state
clusterid	slave	Value	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set ManagerID64 of the core.
clusterpmuirq	master	Signal	DynamiQ pmu irq
CNTHPIRQ	master	Signal	Timer signals to SOC.
CNTHPSIRQ	master	Signal	Timer signals to SOC
CNTHVIRQ	master	Signal	Timer signals to SOC.
CNTHVSIRQ	master	Signal	Timer signals to SOC
CNTPNSIRQ	master	Signal	Timer signals to SOC.
CNTPSIRQ	master	Signal	Timer signals to SOC.
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.
CNTVIRQ	master	Signal	Timer signals to SOC.
commirq	master	Signal	Interrupt signal from debug communications channel.
core_clk_in	slave	ClockSignal	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
core_pcsn_pchannel	master	PChannel	Core PCSM signals
core_powerdown_out	master	Signal	DEPRECATED - An output from PPUs that informs thermal controller of the core power info
core_ppu_hw_stat	master	Value	Notify the power state change inside the PPU.
coreerrirq	master	Signal	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq	master	Signal	Fault indicator for a detected 1 or 2 bit ECC error
coreinstrrun	master	Signal	Core Running State
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
ctidbgirq	master	Signal	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	slave	Signal	External debug interface.
dbgnopwrdown	master	Signal	No power-down request.
dbgprupreq	master	Signal	Debug power up request.
defaultMP	slave	Signal	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main master interface ports.
dev_debug_s	slave	PVBus	External debug interface.

Port	Direction	Protocol	Description
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.
gicreset	master	Signal	An output from PPU that can be used to reset GICCLK domain components
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	slave	PVBus	External debug interface.
pmbirq	master	Signal	Interrupt signal from the statistical profiling unit.
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	master	Signal	PPU cluster interrupt.
ppu_cluster_isolate	master	Signal	PPU Cluster Domain isolation control.
ppu_cluster_wakerequest	slave	Signal	PPU cluster wake request signal.
ppu_core_irq	master	Signal	PPU core interrupt.
ppu_core_isolate	master	Signal	PPU Core Domain isolation control.
ppu_core_wakerequest	slave	Signal	PPU core wake request signal.
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	master	PVBus	The core will generate bus requests on this port.
pvbus_periph_m	master	PVBus	The core can generate peripheral bus request on this port.
rei	slave	Signal	Per core RAM Error Interrupt.
reset	slave	Signal	Reset signal to cluster.
rvbaraddr	slave	Value_64	Reset vector base address.
sci_m	master	SystemCoherencyInterface	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei	slave	Signal	Per core System Error physical pins.
spiden	slave	Signal	External debug interface.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq	master	Signal	Interrupt signal from the trace buffer unit.
utility_bus_s	slave	PVBus	Utility bus slave
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfiq	slave	Signal	Virtualised FIQ.
virq	slave	Signal	Virtualised IRQ.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Per core virtual System Error physical pins.

## Parameters for ARMCortexA725CT\_CortexX925CT

### **AEND0\_DEFAULT**

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

Type: `uint64_t`

Default value: `0x0`

### **AEND1\_DEFAULT**

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

Type: `uint64_t`

Default value: `0x0`

### **AEND2\_DEFAULT**

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

Type: `uint64_t`

Default value: `0x0`

### **AEND3\_DEFAULT**

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

Type: `uint64_t`

Default value: `0x0`

### **ASTART0\_DEFAULT**

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

Type: `uint64_t`

Default value: `0x0`

### **ASTART1\_DEFAULT**

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

Type: `uint64_t`



Default value: 0x0

### **ASTART2\_DEFAULT**

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

Type: `uint64_t`

Default value: 0x0

### **ASTART3\_DEFAULT**

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

Type: `uint64_t`

Default value: 0x0

### **BROADCASTATOMIC**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`

Default value: `true`

### **BROADCASTCACHEMAINT**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

Type: `bool`

Default value: `false`

### **BROADCASTOUTER**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

Type: `bool`

Default value: `false`

### **BROADCASTPERSIST**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: `bool`

Default value: `true`

### **CLUSTER\_ID**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the ManagerID64 of the core.

Type: `uint32_t`

Default value: `0x0`

### **CMO\_broadcast\_when\_cache\_state\_modelling\_disabled**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. This value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required.

Type: `uint8_t`

Default value: `1`

### **GICDISABLE**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`

Default value: `true`

### **bus\_type**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

Type: `uint8_t`

Default value: `0`

### **core\_power\_on\_by\_default**

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPUs which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

Type: `bool`

Default value: `false`

### **`dcache-state_modelled`**

Set whether D-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **`default_opmode`**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

Type: `uint8_t`

Default value: 4

### **`diagnostics`**

Enable DynamIQ diagnostic messages.

Type: `bool`

Default value: `false`

### **`enable_simulation_performance_optimizations`**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of `stage12_tlb_size` parameter to 1024).

Type: `bool`

Default value: `true`

### **`has_peripheral_port`**

If true, an additional AXI peripheral port is configured.

Type: `bool`

Default value: `false`

### **`icache-state_modelled`**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

**l3cache-has\_mpam**

L3 Cache has MPAM support.

Type: `bool`

Default value: `true`

**l3cache-hit\_latency**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-maintenance\_latency**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-miss\_latency**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-read\_access\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size `l3cache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-read\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l3cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **l3cache-size**

L3 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

### **l3cache-snoop\_data\_transfer\_latency**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **l3cache-snoop\_issue\_latency**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **l3cache-ways**

L3 Cache number of ways (sets are implicit from size).

Type: `uint16_t`

Default value: `16`

### **l3cache-write\_access\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size `l3cache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **l3cache-write\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l3cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`mpmm_accumulator_multiplier`**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator. value of `n` means the accumulator will use (`n * accumulator value`) to calculate the mpmm threshold (MPMM). is provided as a fast model workaround to handle cases where execution of quantum in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

Type: `uint8_t`

Default value: `1`

### **`num_acp`**

Number of ACP ports.

Type: `uint8_t`

Default value: `0`

### **`subcluster0.CPUCFR`**

Value of CPU Configuration Register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.NUM_CORES`**

Number of cores in the subcluster. Total number of cores in cluster may not exceed 14.

Type: `uint8_t`

Default value: `1`

### **`subcluster0.cpi_div`**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

### **`subcluster0.cpi_mul`**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: 0x1

#### **subcluster0.cpu0.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: false

#### **subcluster0.cpu0.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: bool

Default value: false

#### **subcluster0.cpu0.CRYPTODISABLE**

Disable cryptographic features.

Type: bool

Default value: false

#### **subcluster0.cpu0.RVBARADDR**

Value of RVBAR\_ELx register.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu0.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter trace\_special\_hlt\_imm16.

Type: bool

Default value: false

#### **subcluster0.cpu0.force-fpsid**

Override the FPSID value.

Type: bool

Default value: true

**subcluster0.cpu0.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t



Default value: 0x100000

**subcluster0.cpu0.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu0.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu0.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster0.cpu0.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster0.cpu0.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: 0x123456

**subcluster0.cpu0.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster0.cpu0.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster0.cpu0.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster0.cpu0.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster0.cpu0.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: true

**subcluster0.cpu0.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu0.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu0.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster0.cpu0.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu0.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu0.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster0.cpu0.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster0.cpu1.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster0.cpu1.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster0.cpu1.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster0.cpu1.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu1.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu1.force-fpsid`**

Override the FPSID value.

Type: `bool`

Default value: `true`

### **`subcluster0.cpu1.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu1.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu1.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu1.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu1.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to

the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu1.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x100000`

#### **`subcluster0.cpu1.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu1.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu1.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu1.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu1.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

**subcluster0.cpu1.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

**subcluster0.cpu1.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu1.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu1.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu1.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster0.cpu1.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster0.cpu1.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu1.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu1.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu1.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu1.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu1.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu1.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`



Default value: 0xf000000

**subcluster0.cpu1.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu1.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster0.cpu1.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: true

**subcluster0.cpu10.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: false

**subcluster0.cpu10.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: bool

Default value: false

**subcluster0.cpu10.CRYPTODISABLE**

Disable cryptographic features.

Type: bool

Default value: false

**subcluster0.cpu10.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu10.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu10.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster0.cpu10.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu10.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu10.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu10.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu10.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu10.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x100000

**subcluster0.cpu10.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu10.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu10.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu10.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu10.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

### **`subcluster0.cpu10.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

### **`subcluster0.cpu10.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu10.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu10.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu10.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster0.cpu10.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster0.cpu10.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster0.cpu10.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster0.cpu10.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu10.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu10.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu10.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu10.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu10.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu10.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster0.cpu10.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster0.cpu11.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster0.cpu11.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster0.cpu11.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster0.cpu11.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu11.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu11.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster0.cpu11.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu11.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu11.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu11.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu11.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu11.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x100000`

### **`subcluster0.cpu11.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`



**subcluster0.cpu11.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu11.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu11.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu11.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu11.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu11.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu11.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu11.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu11.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster0.cpu11.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster0.cpu11.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu11.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu11.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu11.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu11.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu11.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster0.cpu11.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu11.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu11.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster0.cpu11.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster0.cpu12.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu12.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu12.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu12.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu12.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu12.force-fpsid`**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster0.cpu12.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu12.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu12.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu12.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu12.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu12.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x100000

#### **subcluster0.cpu12.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu12.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu12.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu12.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu12.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

#### **subcluster0.cpu12.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu12.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster0.cpu12.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster0.cpu12.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: 0x123456

**subcluster0.cpu12.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster0.cpu12.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster0.cpu12.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster0.cpu12.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster0.cpu12.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: true

**subcluster0.cpu12.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu12.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu12.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster0.cpu12.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu12.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000



**subcluster0.cpu12.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster0.cpu12.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster0.cpu2.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster0.cpu2.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster0.cpu2.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster0.cpu2.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu2.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu2.force-fpsid`**

Override the FPSID value.

Type: `bool`

Default value: `true`

### **`subcluster0.cpu2.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu2.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu2.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu2.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu2.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to

the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu2.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x100000`

#### **`subcluster0.cpu2.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu2.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu2.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu2.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu2.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu2.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu2.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu2.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu2.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu2.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster0.cpu2.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster0.cpu2.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu2.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu2.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu2.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu2.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu2.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu2.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: 0xf000000

**subcluster0.cpu2.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu2.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster0.cpu2.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: true

**subcluster0.cpu3.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: false

**subcluster0.cpu3.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: bool

Default value: false

**subcluster0.cpu3.CRYPTODISABLE**

Disable cryptographic features.

Type: bool

Default value: false

**subcluster0.cpu3.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu3.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu3.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster0.cpu3.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu3.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu3.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu3.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x100000

**subcluster0.cpu3.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu3.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.



Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu3.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu3.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

### **`subcluster0.cpu3.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

### **`subcluster0.cpu3.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu3.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu3.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu3.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster0.cpu3.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster0.cpu3.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster0.cpu3.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster0.cpu3.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu3.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu3.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu3.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu3.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu3.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu3.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster0.cpu3.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster0.cpu4.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster0.cpu4.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster0.cpu4.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster0.cpu4.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu4.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu4.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster0.cpu4.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu4.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu4.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu4.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu4.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu4.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x100000`

### **`subcluster0.cpu4.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu4.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu4.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu4.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu4.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu4.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu4.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster0.cpu4.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster0.cpu4.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu4.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu4.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu4.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu4.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu4.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster0.cpu4.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu4.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu4.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster0.cpu4.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).



Type: `bool`

Default value: `true`

#### **`subcluster0.cpu5.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu5.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu5.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu5.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu5.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

#### **`subcluster0.cpu5.force-fpsid`**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster0.cpu5.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x100000

#### **subcluster0.cpu5.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu5.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu5.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu5.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster0.cpu5.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

#### **subcluster0.cpu5.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu5.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu5.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu5.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu5.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster0.cpu5.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster0.cpu5.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster0.cpu5.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster0.cpu5.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: true

**subcluster0.cpu5.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu5.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu5.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster0.cpu5.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu5.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu5.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster0.cpu5.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster0.cpu6.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster0.cpu6.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster0.cpu6.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster0.cpu6.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu6.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu6.force-fpsid`**

Override the FPSID value.

Type: `bool`

Default value: `true`

### **`subcluster0.cpu6.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu6.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu6.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu6.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu6.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to

the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu6.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x100000`

#### **`subcluster0.cpu6.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu6.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu6.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster0.cpu6.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`



**subcluster0.cpu6.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu6.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu6.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu6.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu6.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster0.cpu6.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster0.cpu6.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster0.cpu6.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu6.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu6.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu6.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu6.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu6.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu6.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: 0xf000000

**subcluster0.cpu6.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu6.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster0.cpu6.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: true

**subcluster0.cpu7.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: false

**subcluster0.cpu7.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: bool

Default value: false

**subcluster0.cpu7.CRYPTODISABLE**

Disable cryptographic features.

Type: bool

Default value: false

**subcluster0.cpu7.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu7.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu7.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster0.cpu7.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu7.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu7.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu7.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x100000

**subcluster0.cpu7.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu7.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu7.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu7.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

### **`subcluster0.cpu7.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

### **`subcluster0.cpu7.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu7.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster0.cpu7.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu7.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster0.cpu7.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster0.cpu7.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster0.cpu7.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster0.cpu7.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu7.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: 0x0

**subcluster0.cpu7.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: 0xf000000

**subcluster0.cpu7.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster0.cpu7.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster0.cpu7.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu7.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster0.cpu7.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster0.cpu8.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`



**subcluster0.cpu8.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster0.cpu8.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster0.cpu8.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu8.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster0.cpu8.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster0.cpu8.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu8.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu8.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu8.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu8.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu8.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x100000`

### **`subcluster0.cpu8.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu8.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu8.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster0.cpu8.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu8.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu8.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu8.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster0.cpu8.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster0.cpu8.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster0.cpu8.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster0.cpu8.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster0.cpu8.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu8.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu8.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster0.cpu8.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu8.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu8.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster0.cpu8.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster0.cpu9.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster0.cpu9.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu9.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu9.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.cpu9.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster0.cpu9.force-fpsid`**

Override the FPSID value.

Type: `bool`

Default value: `true`

**subcluster0.cpu9.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu9.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu9.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu9.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu9.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.cpu9.l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: 0x100000

**subcluster0.cpu9.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster0.cpu9.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t



Default value: 0

**subcluster0.cpu9.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu9.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster0.cpu9.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster0.cpu9.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster0.cpu9.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster0.cpu9.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster0.cpu9.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster0.cpu9.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: true

**subcluster0.cpu9.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster0.cpu9.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu9.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster0.cpu9.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster0.cpu9.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster0.cpu9.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster0.cpu9.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster0.dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.dcache-maintenance\_latency**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.dcache-miss\_latency**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster0.dcache-prefetch\_enabled**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

Type: `bool`

Default value: `false`

**subcluster0.dcache-read\_access\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.dcache-size**

L1 D-Cache size in bytes.

Type: uint32\_t

Default value: 0x10000

**subcluster0.dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.dcache-write\_access\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.ete.CLAIMTAGS**

Number of claim tags.

Type: uint8\_t

Default value: 4

**subcluster0.ete.ETE\_REVISION**

ETE revision: 0=ETEv1.0, 1=ETEv1.1, 2=ETEv1.2, 3=ETEv1.3.

Type: uint8\_t

Default value: 1

**subcluster0.ete.MAX\_INST\_PER\_Q**

Maximum limit for the number of instructions implied by a Q element.

Type: uint16\_t

Default value: 0x1

**subcluster0.ete.NumberOfRSPairs**

Number of resource selector pairs.

Type: uint8\_t

Default value: 8

**subcluster0.ete.PIDR\_CM0D**

TRCPIDR CM0D value.

Type: uint8\_t

Default value: 0

**subcluster0.ete.PIDR\_REVAND**

TRCPIDR REVAND value.

Type: `uint8_t`

Default value: 0

#### **`subcluster0.ete.PIDR_REVISION`**

TRCPIDR REVISION value.

Type: `uint8_t`

Default value: 0

#### **`subcluster0.ete.Q_CADENCE`**

Number of instruction blocks traced between two Q elements.

Type: `uint16_t`

Default value: 0x1

#### **`subcluster0.ete.RES0_STATEFUL`**

Whether **RES0** bits are stateful or **RAZ/WI**.

Type: `bool`

Default value: `false`

#### **`subcluster0.ete.RETSTACK`**

Return stack depth.

Type: `uint8_t`

Default value: 1

#### **`subcluster0.ete.REVISION`**

TRCIDR1 revision value.

Type: `uint8_t`

Default value: 0

#### **`subcluster0.ete.SIM_OVERFLOW_GRANULARITY`**

Number of instruction blocks in each granule, for simulated overflow.

Type: `uint32_t`

Default value: 0x64

#### **`subcluster0.ete.SIM_OVERFLOW_PERCENTAGE`**

Percentage of instruction blocks lost in each granule, for simulated overflow.

Type: `uint8_t`

Default value: 0

### **`subcluster0.ete.SOURCE_ADDRESS`**

Allow generation of source address elements.

Type: `bool`

Default value: `false`

### **`subcluster0.ete.TRACE_OUTPUT`**

File to which to write trace byte stream.

Type: `string`

Default value: N/A

### **`subcluster0.ete.TRCRSRTA_FORCED_EXCEP`**

TRCRSR.TA value for a forcibly traced exception.

Type: `bool`

Default value: `false`

### **`subcluster0.ete.TSMARK`**

Whether timestamp markers are supported.

Type: `bool`

Default value: `true`

### **`subcluster0.force_mte_tag_access_razwi_and_ignore_tag_checks`**

Force MTE tag accesses to **RAZ/WI** and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

Type: `bool`

Default value: `false`

### **`subcluster0.force_zero_PSTATE_PAN`**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

### **subcluster0.force\_zero\_mpam\_partid\_and\_pmg**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAMO\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

Type: `uint8_t`

Default value: `0`

### **subcluster0.has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

### **subcluster0.has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

### **subcluster0.has\_enhanced\_pan**

Implements Armv8.7 Enhanced PAN feature (FEAT\_PAN3) values of this parameter are:- 1, feature is implemented if ARMv8.7 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: `2`

### **subcluster0.has\_ete**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (-plugin or -P).

Type: `bool`

Default value: `false`

### **subcluster0.has\_large\_va**

Implement support for the extended 52-bit virtual addresses from ARMv8.2 (FEAT\_LVA). values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.2 is enabled.- 2, feature is implemented.

Type: `uint8_t`



Default value: 0

**subcluster0.has\_statistical\_profiling**

Whether Statistical Based Profiling is implemented (FEAT\_SPE).

Type: bool

Default value: false

**subcluster0.icache-hit\_latency**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster0.icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

Type: bool

Default value: false

**subcluster0.icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.icache-read_latency`**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster0.icache-size`**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: `0x10000`

### **`subcluster0.instruction_tlb_size`**

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

Type: `uint32_t`

Default value: `0x0`

### **`subcluster0.invalidate_code_cache_on_icache_cmo`**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: `uint8_t`

Default value: `0`

### **`subcluster0.memory_tagging_support_level`**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

Type: `uint8_t`

Default value: `3`

### **`subcluster0.pmu-num_counters`**

Number of PMU counters implemented.

Type: `uint8_t`

Default value: 6

### **`subcluster0.ptw_latency`**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

### **`subcluster0.stage12_tlb_size`**

Number of stage1+2 tlb entries. Valid values are 0 or  $\geq 4$ .

Type: `uint32_t`

Default value: 0x80

### **`subcluster0.tlb_latency`**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

### **`subcluster0.tlbi_stall_enabled`**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: `bool`

Default value: `false`

### **`subcluster0.treat_PAC_as_NOP`**

Non-architecture parameter. Treat Pointer Authentication as **NOP**. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are **NOP**. Branch, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

Type: `bool`

Default value: `false`

### **`subcluster0.walk_cache_latency`**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

**subcluster1.CPUCFR**

Value of CPU Configuration Register.

Type: uint64\_t

Default value: 0x0

**subcluster1.NUM\_CORES**

Number of cores in the subcluster. Total number of cores in cluster may not exceed 14.

Type: uint8\_t

Default value: 1

**subcluster1.cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: uint32\_t

Default value: 0x1

**subcluster1.cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: uint32\_t

Default value: 0x1

**subcluster1.cpu0.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: false

**subcluster1.cpu0.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: bool

Default value: false

**subcluster1.cpu0.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu0.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu0.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu0.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu0.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu0.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu0.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu0.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu0.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

### **`subcluster1.cpu0.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu0.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu0.l2cache-ways`**

L2 Cache number of ways (sets are implicit from size).

Type: `uint8_t`

Default value: `8`

### **`subcluster1.cpu0.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu0.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu0.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

### **`subcluster1.cpu0.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

### **`subcluster1.cpu0.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu0.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu0.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu0.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster1.cpu0.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster1.cpu0.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster1.cpu0.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster1.cpu0.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu0.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu0.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`



**subcluster1.cpu0.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu0.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu0.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu0.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu0.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster1.cpu1.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu1.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster1.cpu1.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu1.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu1.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu1.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu1.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu1.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu1.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu1.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu1.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

#### **`subcluster1.cpu1.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu1.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu1.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster1.cpu1.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu1.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu1.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu1.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu1.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu1.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu1.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu1.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster1.cpu1.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster1.cpu1.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu1.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu1.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu1.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu1.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu1.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster1.cpu1.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu1.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu1.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster1.cpu1.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster1.cpu10.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu10.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu10.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu10.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu10.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu10.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu10.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu10.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu10.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu10.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu10.l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

**subcluster1.cpu10.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`



Default value: 0x0

**subcluster1.cpu10.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu10.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster1.cpu10.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu10.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu10.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu10.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu10.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster1.cpu10.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster1.cpu10.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: 0x123456

**subcluster1.cpu10.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster1.cpu10.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster1.cpu10.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster1.cpu10.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster1.cpu10.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: true

**subcluster1.cpu10.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu10.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu10.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster1.cpu10.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu10.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu10.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu10.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster1.cpu11.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu11.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster1.cpu11.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu11.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu11.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu11.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu11.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu11.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu11.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu11.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu11.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster1.cpu11.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu11.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu11.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster1.cpu11.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu11.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu11.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu11.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu11.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu11.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu11.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu11.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster1.cpu11.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster1.cpu11.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu11.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu11.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu11.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu11.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu11.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu11.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`



Default value: 0xf000000

**subcluster1.cpu11.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu11.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster1.cpu11.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: true

**subcluster1.cpu12.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: false

**subcluster1.cpu12.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: bool

Default value: false

**subcluster1.cpu12.CRYPTODISABLE**

Disable cryptographic features.

Type: bool

Default value: false

**subcluster1.cpu12.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu12.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu12.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu12.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu12.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu12.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu12.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu12.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster1.cpu12.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu12.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu12.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster1.cpu12.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu12.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu12.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu12.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu12.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu12.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu12.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu12.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

### **`subcluster1.cpu12.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

### **`subcluster1.cpu12.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **`subcluster1.cpu12.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **`subcluster1.cpu12.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster1.cpu12.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: 0x0

### **`subcluster1.cpu12.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: 0xf000000

### **`subcluster1.cpu12.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster1.cpu12.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster1.cpu12.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`subcluster1.cpu12.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu12.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster1.cpu2.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster1.cpu2.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu2.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu2.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu2.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu2.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu2.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu2.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu2.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu2.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster1.cpu2.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu2.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu2.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t



Default value: 8

**subcluster1.cpu2.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu2.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu2.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu2.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu2.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu2.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu2.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu2.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster1.cpu2.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster1.cpu2.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu2.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu2.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu2.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu2.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu2.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu2.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu2.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu2.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu2.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster1.cpu3.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu3.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster1.cpu3.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu3.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu3.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu3.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu3.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu3.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu3.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

### **`subcluster1.cpu3.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster1.cpu3.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu3.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu3.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu3.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu3.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu3.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu3.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster1.cpu3.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster1.cpu3.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu3.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu3.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu3.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu3.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu3.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu3.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu3.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu3.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.



Type: `bool`

Default value: `false`

### **`subcluster1.cpu3.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster1.cpu4.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster1.cpu4.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu4.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu4.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu4.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu4.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu4.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu4.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu4.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu4.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu4.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster1.cpu4.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu4.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu4.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster1.cpu4.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu4.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu4.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu4.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu4.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu4.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu4.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu4.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster1.cpu4.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster1.cpu4.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**subcluster1.cpu4.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster1.cpu4.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu4.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu4.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu4.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu4.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu4.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster1.cpu4.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu4.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster1.cpu5.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu5.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu5.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu5.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu5.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter trace\_special\_hlt\_imm16.

Type: `bool`

Default value: `false`

**subcluster1.cpu5.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu5.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu5.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu5.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu5.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: 0x0

### **subcluster1.cpu5.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

### **subcluster1.cpu5.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster1.cpu5.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster1.cpu5.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

### **subcluster1.cpu5.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster1.cpu5.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t



Default value: 0x0

**subcluster1.cpu5.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu5.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu5.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu5.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu5.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu5.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster1.cpu5.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

### **`subcluster1.cpu5.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **`subcluster1.cpu5.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **`subcluster1.cpu5.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`subcluster1.cpu5.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu5.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`subcluster1.cpu5.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`subcluster1.cpu5.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

#### **`subcluster1.cpu5.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`subcluster1.cpu5.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu5.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`subcluster1.cpu6.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu6.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

#### **`subcluster1.cpu6.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu6.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu6.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu6.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu6.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu6.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu6.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

#### **subcluster1.cpu6.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu6.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

#### **subcluster1.cpu6.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu6.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.cpu6.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

#### **subcluster1.cpu6.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster1.cpu6.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster1.cpu6.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

### **subcluster1.cpu6.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

### **subcluster1.cpu6.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

### **subcluster1.cpu6.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

### **subcluster1.cpu6.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu6.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster1.cpu6.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster1.cpu6.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster1.cpu6.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster1.cpu6.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu6.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: 0x0

**subcluster1.cpu6.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: 0xf000000

**subcluster1.cpu6.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu6.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu6.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu6.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu6.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster1.cpu7.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`



**subcluster1.cpu7.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster1.cpu7.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu7.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu7.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**subcluster1.cpu7.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu7.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu7.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu7.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu7.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu7.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

#### **`subcluster1.cpu7.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.cpu7.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu7.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster1.cpu7.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu7.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu7.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu7.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu7.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu7.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**subcluster1.cpu7.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**subcluster1.cpu7.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**subcluster1.cpu7.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**subcluster1.cpu7.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu7.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu7.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu7.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu7.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu7.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster1.cpu7.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu7.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu7.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**subcluster1.cpu7.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`subcluster1.cpu8.CFGEND`**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **`subcluster1.cpu8.CFGTE`**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu8.CRYPTODISABLE`**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu8.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu8.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu8.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu8.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu8.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu8.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu8.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu8.l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

**subcluster1.cpu8.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**subcluster1.cpu8.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu8.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster1.cpu8.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu8.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu8.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu8.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t



Default value: 0

**subcluster1.cpu8.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster1.cpu8.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**subcluster1.cpu8.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: 0x123456

**subcluster1.cpu8.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**subcluster1.cpu8.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**subcluster1.cpu8.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**subcluster1.cpu8.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**subcluster1.cpu8.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: true

**subcluster1.cpu8.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu8.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu8.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**subcluster1.cpu8.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**subcluster1.cpu8.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu8.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**subcluster1.cpu8.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**subcluster1.cpu9.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

**subcluster1.cpu9.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**subcluster1.cpu9.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**subcluster1.cpu9.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu9.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`subcluster1.cpu9.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu9.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu9.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu9.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`subcluster1.cpu9.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu9.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**subcluster1.cpu9.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**subcluster1.cpu9.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**subcluster1.cpu9.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**subcluster1.cpu9.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**subcluster1.cpu9.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu9.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**subcluster1.cpu9.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**subcluster1.cpu9.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**subcluster1.cpu9.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**subcluster1.cpu9.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**subcluster1.cpu9.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**subcluster1.cpu9.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**subcluster1.cpu9.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.cpu9.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**subcluster1.cpu9.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**subcluster1.cpu9.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: 0xf000000

#### **subcluster1.cpu9.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

#### **subcluster1.cpu9.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

#### **subcluster1.cpu9.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: true

#### **subcluster1.dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.dcache-maintenance\_latency**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

#### **subcluster1.dcache-miss\_latency**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t



Default value: 0x0

### **subcluster1.dcache-prefetch\_enabled**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

Type: bool

Default value: false

### **subcluster1.dcache-read\_access\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster1.dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster1.dcache-size**

L1 D-Cache size in bytes.

Type: uint32\_t

Default value: 0x10000

### **subcluster1.dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **subcluster1.dcache-write\_access\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be

used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.dcache-write_latency`**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.ecv_support_level`**

Implement Enhanced Counter Virtualization feature from ARMv8.6. 0, Not supported. 1, fully supported without CNTPOFF. 2, fully supported with CNTPOFF (FEAT\_ECV).

Type: `uint8_t`

Default value: 2

#### **`subcluster1.ete.CLAIMTAGS`**

Number of claim tags.

Type: `uint8_t`

Default value: 4

#### **`subcluster1.ete.MAX_INST_PER_Q`**

Maximum limit for the number of instructions implied by a Q element.

Type: `uint16_t`

Default value: `0x1`

#### **`subcluster1.ete.NumberOfRSPairs`**

Number of resource selector pairs.

Type: `uint8_t`

Default value: 8

#### **`subcluster1.ete.PIDR_CM0D`**

TRCPIDR CM0D value.

Type: `uint8_t`

Default value: 0

#### **`subcluster1.ete.Q_CADENCE`**

Number of instruction blocks traced between two Q elements.

Type: `uint16_t`

Default value: 0x1

#### **`subcluster1.ete.RES0_STATEFUL`**

Whether **RES0** bits are stateful or **RAZ/WI**.

Type: `bool`

Default value: `false`

#### **`subcluster1.ete.RETSTACK`**

Return stack depth.

Type: `uint8_t`

Default value: 3

#### **`subcluster1.ete.SIM_OVERFLOW_GRANULARITY`**

Number of instruction blocks in each granule, for simulated overflow.

Type: `uint32_t`

Default value: 0x64

#### **`subcluster1.ete.SIM_OVERFLOW_PERCENTAGE`**

Percentage of instruction blocks lost in each granule, for simulated overflow.

Type: `uint8_t`

Default value: 0

#### **`subcluster1.ete.SOURCE_ADDRESS`**

Allow generation of source address elements.

Type: `bool`

Default value: `false`

#### **`subcluster1.ete.TRACE_OUTPUT`**

File to which to write trace byte stream.

Type: `string`

Default value: `N/A`

#### **`subcluster1.ete.TRCRSRTA_FORCED_EXCEP`**

TRCRSR.TA value for a forcibly traced exception.

Type: `bool`

Default value: `false`

#### **`subcluster1.ext_abort_so_write_ras_type`**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO , 4 = UER, 5 = CE.

Type: `uint8_t`

Default value: `2`

#### **`subcluster1.force_mte_tag_access_razwi_and_ignore_tag_checks`**

Force MTE tag accesses to **RAZ/WI** and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

Type: `bool`

Default value: `false`

#### **`subcluster1.force_zero_PSTATE_PAN`**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

#### **`subcluster1.force_zero_mpam_partid_and_pmg`**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAMO\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

Type: `uint8_t`

Default value: `0`

#### **`subcluster1.has_delayed_dbgreg`**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

### **`subcluster1.has_delayed_sysreg`**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

### **`subcluster1.has_enhanced_pan`**

Implements Armv8.7 Enhanced PAN feature (FEAT\_PAN3) values of this parameter are:- 1, feature is implemented if ARMv8.7 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 2

### **`subcluster1.has_ete`**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (-plugin or -P).

Type: `bool`

Default value: `false`

### **`subcluster1.has_mt_pmu_disable_feature`**

Implements Multi-threading PMU disable extension from ARMv8.6 (FEAT\_MTPMU). 0: FEAT\_MTPMU is disabled, 1: FEAT\_MTPMU is enabled if ARMv8.6 is implemented, 2: FEAT\_MTPMU is cherry-picked, 0xF: The feature is disabled and is represented by value 0xF in ID\_AA64DFR0\_EL1.MTPMU.

Type: `uint8_t`

Default value: 0

### **`subcluster1.has_statistical_profiling`**

Whether Statistical Based Profiling is implemented (FEAT\_SPE).

Type: `bool`

Default value: `true`

### **`subcluster1.has_v8_7_spe_inverted_filtering`**

Where FEAT\_SPEv1p2 is implemented, whether inverted filtering by events is implemented (represented by PMISDR.FnE).

Type: `bool`

Default value: `false`

### **subcluster1.has\_v8\_7\_spe\_previous\_branch\_target**

Where FEAT\_SPEv1p2 is implemented, whether the optional branch target feature is implemented (FEAT\_SPE\_PBT).

Type: `bool`

Default value: `false`

### **subcluster1.icache-hit\_latency**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **subcluster1.icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **subcluster1.icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **subcluster1.icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

### **subcluster1.icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and

intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.icache-read_latency`**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

#### **`subcluster1.icache-size`**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: `0x10000`

#### **`subcluster1.instruction_tlb_size`**

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

Type: `uint32_t`

Default value: `0x0`

#### **`subcluster1.invalidate_code_cache_on_icache_cmo`**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: `uint8_t`

Default value: `0`

#### **`subcluster1.memory_tagging_support_level`**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

Type: `uint8_t`

Default value: `3`

**subcluster1.mpam\_has\_altsp**

DEPRECATED: MPAMIDR\_EL1.HAS\_ALTSP is required when FEAT\_RME is implemented.

Type: `bool`

Default value: `false`

**subcluster1.mpamidr\_has\_force\_ns**

Whether MPAMIDR\_EL1.HAS\_FORCE\_NS bit is set or clear. values of this parameter are:-  
0, feature is not enabled.- 1, feature is implemented if ARMv8.6 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

**subcluster1.mpamidr\_has\_sdeflt**

Whether MPAMIDR\_EL1.HAS\_SDEFLT bit is set or clear. values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.6 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

**subcluster1.mpamidr\_has\_tidr**

Whether MPAMIDR\_EL1.HAS\_TIDR bit is set or clear. values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.6 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

**subcluster1.pmu-num\_counters**

Number of PMU counters implemented.

Type: `uint8_t`

Default value: 31

**subcluster1.ptw\_latency**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0



**subcluster1.stage12\_tlb\_size**

Number of stage1+2 tlb entries. Valid values are 0 or  $\geq 4$ .

Type: `uint32_t`

Default value: `0x80`

**subcluster1.tcr\_txsz\_undersize\_should\_fault**

If large VA is not supported, Whether undersized TxSZ value should generate translation fault.

Type: `bool`

Default value: `false`

**subcluster1.tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

**subcluster1.tlbi\_stall\_enabled**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: `bool`

Default value: `false`

**subcluster1.treat\_PAC\_as\_NOP**

Non-architecture parameter. Treat Pointer Authentication as **NOP**. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are **NOP**. Branch, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

Type: `bool`

Default value: `false`

**subcluster1.walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

## 3.70 ARMCortexA72CT

Defined in `LISA/ARMCortexA72CT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following parameters were added:

- `has_delayed_dbgreg`
- `has_delayed_sysreg`

### About ARMCortexA72CT

The per-core parameters are preceded by `cpun.`, where `n` identifies the core (0-3).

- The cache latency cluster parameters are only effective when you enable cache-state modeling.
- Timing annotation for transactions downstream of the cache and TLB models propagates through the cache and TLB models.
- ECC and parity schemes are hardware-specific so are not supported.
- This model has a variable number of cores per cluster, specified using the `NUM_CORES` parameter.
- This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (`crypto.dll` or `crypto.so`) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

### Iris and MTI instances for ARMCortexA72CT

This model has the following Iris instances:

Name	Instance type
ARMCortexA72CT	Cluster_ARM_Cortex-A72
ARMCortexA72CT.AMU	PVBusLogger
ARMCortexA72CT.AMU.mapper	PVBusMapper
ARMCortexA72CT.DAP	PVBusLogger
ARMCortexA72CT.DAP.mapper	PVBusMapper
ARMCortexA72CT.DSU	DSU
ARMCortexA72CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA72CT.MMAP	PVBusLogger
ARMCortexA72CT.MMAP.mapper	PVBusMapper
ARMCortexA72CT.RAS	PVBusLogger
ARMCortexA72CT.RAS.mapper	PVBusMapper

Name	Instance type
ARMCortexA72CT.acp_mapper	PVBusMapper
ARMCortexA72CT.cpu0	ARM_Cortex-A72
ARMCortexA72CT.cpu0.UTLB	TLB
ARMCortexA72CT.cpu0.dtlb	TLB
ARMCortexA72CT.cpu0.l1dcache	PVCache
ARMCortexA72CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA72CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA72CT.cpu0.l1licache	PVCache
ARMCortexA72CT.cpu0.l1licache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA72CT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexA72CT.ext_bus	PVBusLogger
ARMCortexA72CT.ext_bus.mapper	PVBusMapper
ARMCortexA72CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA72CT.global_debug_rom	debug_rom
ARMCortexA72CT.l2_cache	PVCache
ARMCortexA72CT.l2_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA72CT.l2_cache.upstream[Z] (where Z = 0-16)	PVBusSlave
ARMCortexA72CT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

Name	Component type
ARMCortexA72CT.AMU	PVBusLogger
ARMCortexA72CT.AMU.mapper	PVBusMapper
ARMCortexA72CT.DAP	PVBusLogger
ARMCortexA72CT.DAP.mapper	PVBusMapper
ARMCortexA72CT.DSU	DSU
ARMCortexA72CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA72CT.MMAP	PVBusLogger
ARMCortexA72CT.MMAP.mapper	PVBusMapper
ARMCortexA72CT.RAS	PVBusLogger
ARMCortexA72CT.RAS.mapper	PVBusMapper
ARMCortexA72CT.acp_mapper	PVBusMapper
ARMCortexA72CT.cpu0	ARM_Cortex-A72
ARMCortexA72CT.cpu0.UTLB	TLB
ARMCortexA72CT.cpu0.l1dcache	PVCache
ARMCortexA72CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA72CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA72CT.cpu0.l1licache	PVCache
ARMCortexA72CT.cpu0.l1licache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA72CT.cpu0.l1licache.upstream[0]	PVBusSlave

Name	Component type
ARMCortexA72CT.ext_bus	PVBusLogger
ARMCortexA72CT.ext_bus.mapper	PVBusMapper
ARMCortexA72CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA72CT.l2_cache	PVCache
ARMCortexA72CT.l2_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA72CT.l2_cache.upstream[Z] (where Z = 0-16)	PVBusSlave
ARMCortexA72CT.l2_flusher	AsyncCacheFlushUnit

## Ports for ARMCortexA72CT

Port	Direction	Protocol	Description
aa64naa32	slave	Signal	Register width after reset.
acp_s	slave	PVBus	AXI ACP slave port.
broadcastcachemaint	slave	Signal	Enable broadcasting of cache maintenance operations to downstream caches.
broadcastinner	slave	Signal	Enable broadcasting of Inner Shareable transactions.
broadcastouter	slave	Signal	Enable broadcasting of Outer Shareable transactions.
cfgend	slave	Signal	This signal is for EE bit initialisation.
cfgsdisable	slave	Signal	This signal disables write access to some secure Interrupt Controller registers.
cfgte	slave	Signal	This signal provides default exception handling state.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
clrexmonack	master	Signal	Acknowledge handshake signal for the clrexmonreq signal
clrexmonreq	slave	Signal	Signals the clearing of an external global exclusive monitor
clusterid	slave	Value	The port sets the value of the affinity levels 1 and 2; bits [23:16] and [15:8] of the MPIDR.
CNTHPIRQ	master	Signal	The per-EL counter signal.
CNTPNSIRQ	master	Signal	The per-EL counter signal.
CNTPSIRQ	master	Signal	The per-EL counter signal.
cntvalueb	slave	CounterInterface	Interface to SoC level counter module
CNTVIRQ	master	Signal	The per-EL counter signal.
commirq	master	Signal	Interrupt signal from debug communications channel.
commr_x	master	Signal	Receive portion of Data Transfer Register full.
commt_x	master	Signal	Transmit portion of Data Transfer Register empty.
cp15sdisable	slave	Signal	This signal disables write access to some system control processor registers.
cpuporeset	slave	Signal	Power on reset. Initializes all the processor logic, including debug logic.
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.

Port	Direction	Protocol	Description
ctidbgirq	master	Signal	Cross Trigger Interface (CTI) interrupt trigger output.
dbgack	master	Signal	External debug interface.
dbgen	slave	Signal	External debug interface.
dbgnopwrdown	master	Signal	These signals relate to core power down.
dbgpwrupreq	master	Signal	These signals relate to core power down.
dev_debug_s	slave	PVBus	External debug interface.
edbgrq	slave	Signal	External debug interface.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
l2flushdone	master	Signal	Flush of L2 memory system complete.
l2flushreq	slave	Signal	Request flush of L2 memory system.
l2reset	slave	Signal	Reset the shared L2 memory system controller.
memorymapped_debug_s	slave	PVBus	External debug interface.
niden	slave	Signal	External debug interface.
periphbase	slave	Value_64	This port sets the base address of private peripheral region.
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	master	PVBus	The core will generate bus requests on this port.
rei	slave	Signal	Individual processor RAM Error Interrupt signal input.
reset	slave	Signal	Raising this signal will put the core into reset mode.
romaddr	slave	Value_64	Debug ROM base address.
romaddrv	slave	Signal	Debug ROM base address valid.
rvbaraddr	slave	Value_64	Reset vector base address.
sei	slave	Signal	Per core System Error physical pins.
smpen	master	Signal	This signals AMP or SMP mode for each core.
spiden	slave	Signal	External debug interface.
spniden	slave	Signal	External debug interface.
standbywfe	master	Signal	This signal indicates if a core is in WFE state.
standbywfi	master	Signal	This signal indicates if a core is in WFI state.
standbywfi12	master	Signal	Indicate that all the individual processors and the L2 systems are in a WFI state.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfiq	slave	Signal	Virtual FIQ

Port	Direction	Protocol	Description
vinithi	slave	Signal	This signal controls of the location of the exception vectors at reset.
virq	slave	Signal	Virtual IRQ.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Processor Virtual System Error Interrupt request.

## Parameters for ARMCortexA72CT

### **cpuX.AA64nAA32**

Type: `bool`

Default value: `true`

### **cpuX.CFGEND**

Type: `bool`

Default value: `false`

### **cpuX.CFGTE**

Type: `bool`

Default value: `false`

### **cpuX.CP15SDISABLE**

Type: `bool`

Default value: `false`

### **cpuX.CRYPTODISABLE**

Type: `bool`

Default value: `false`

### **cpuX.RVBARADDR**

Type: `uint64_t`

Default value: `0`

### **cpuX.VINITHI**

Type: `bool`

Default value: `false`

**cpuX.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter trace\_special\_hlt\_imm16.

Type: bool

Default value: false

**cpuX.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint64\_t

Default value: 256

**cpuX.min\_sync\_level**

force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint32\_t

Default value: 0

**cpuX.semihosting-A32\_HLT**

Type: uint32\_t

Default value: 0xF000

**cpuX.semihosting-A64\_HLT**

Type: uint32\_t

Default value: 0xF000

**cpuX.semihosting-ARM\_SVC**

Type: uint32\_t

Default value: 0x123456

**cpuX.semihosting-T32\_HLT**

Type: uint32\_t

Default value: 0x3c

**cpuX.semihosting-Thumb\_SVC**

Type: uint32\_t

Default value: 0xAB

**cpuX.semihosting-cmd\_line**

Type: string

Default value: ""

**cpuX.semihosting-cwd**

Type: string

Default value: ""

**cpuX.semihosting-enable**

Type: bool

Default value: true

**cpuX.semihosting-heap\_base**

Type: uint32\_t

Default value: 0x0

**cpuX.semihosting-heap\_limit**

Type: uint32\_t

Default value: 0x0F000000

**cpuX.semihosting-stack\_base**

Type: uint32\_t

Default value: 0x10000000

**cpuX.semihosting-stack\_limit**

Type: uint32\_t

Default value: 0x0F000000

**cpuX.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**cpuX.vfp-enable\_at\_reset**

Type: bool

Default value: false



**BROADCASTCACHEMAINT**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

Type: `bool`

Default value: `true`

**BROADCASTINNER**

Enable broadcasting of Inner Shareable transactions. The broadcastinner signal will override this value if used.

Type: `bool`

Default value: `true`

**BROADCASTOUTER**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

Type: `bool`

Default value: `true`

**CLUSTER\_ID**

Type: `uint32_t`

Default value: 0

**DBGROMADDR**

Type: `uint64_t`

Default value: `0x22000000`

**DBGROMADDRV**

Type: `bool`

Default value: `true`

**GICDISABLE**

Type: `bool`

Default value: `true`

**NUM\_CORES**

Number of cores in cluster.

Type: `uint8_t`

Default value: 1

### **PERIPHBASE**

Type: `uint64_t`

Default value: 0x13080000

### **bus\_type**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

Type: `uint32_t`

Default value: 0

### **cpi\_div**

divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: 1

### **cpi\_mul**

multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: 1

### **dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit.

Type: `uint64_t`

Default value: 0

### **dcache-maintenance\_latency**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

### **dcache-miss\_latency**

L1 D-Cache timing annotation latency for miss.

Type: `uint64_t`

Default value: 0

### **`dcache-read_access_latency`**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (based on the size of `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

### **`dcache-read_latency`**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

### **`dcache-snoop_data_transfer_latency`**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

### **`dcache-state_modelled`**

Type: `bool`

Default value: `false`

### **`dcache-write_access_latency`**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (based on the size of `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

### **`dcache-write_latency`**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

### **has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

### **has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

### **icache-hit\_latency**

L1 I-Cache timing annotation latency for hit.

Type: `uint64_t`

Default value: 0

### **icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

### **icache-miss\_latency**

L1 I-Cache timing annotation latency for miss.

Type: `uint64_t`

Default value: 0

### **icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

**icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (based on the size of icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0

**icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. This is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0

**icache-state\_modelled**

Type: bool

Default value: false

**l2cache-hit\_latency**

L2 Cache timing annotation latency for hit.

Type: uint64\_t

Default value: 0

**l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0

**l2cache-miss\_latency**

L2 Cache timing annotation latency for miss.

Type: uint64\_t

Default value: 0

**l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access (based on the size of l2cache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0

**l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0

**l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0

**l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0

**l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access (based on the size of l2cache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0

### **l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

### **ptw\_latency**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint32_t`

Default value: 0

### **tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint32_t`

Default value: 0

## 3.71 ARMCortexA73CT

Defined in `LISA/ARMCortexA73CT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following parameters were added:

- `has_delayed_dbgreg`
- `has_delayed_sysreg`

### About ARMCortexA73CT

- The per-core parameters are preceded by `cpun.`, where `n` identifies the core (0-3).

- This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (`crypto.dll` or `crypto.so`) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).
- If either L1 cache is stateful, then the L2 cache is stateful. This is controlled by the `dcache-state_modelled` and `icache-state_modelled` parameters.
- The `semihosting-cwd` parameter sets the CWD that is used for semihosting. The host operating system limits the maximum path length. The `semihosting-cwd` parameter does not provide any security. Software running on the model can access files outside this directory using relative paths containing `..` or using absolute paths.

This model has a variable number of cores per cluster, specified using the `NUM_CORES` parameter.

- The system designer decides whether a debug APB ties the external debug view with other system views. In the model, use a `PVBusDecoder` to direct traffic to the correct port, `dev_debug_s` or `memorymapped_debug_s`.
- ECC support is hardware-specific so is not modeled.

## Iris and MTI instances for ARM Cortex A73CT

This model has the following Iris instances:

Name	Instance type
ARMCortexA73CT	Cluster_ARM_Cortex-A73
ARMCortexA73CT.AMU	PVBusLogger
ARMCortexA73CT.AMU.mapper	PVBusMapper
ARMCortexA73CT.DAP	PVBusLogger
ARMCortexA73CT.DAP.mapper	PVBusMapper
ARMCortexA73CT.DSU	DSU
ARMCortexA73CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA73CT.MMAP	PVBusLogger
ARMCortexA73CT.MMAP.mapper	PVBusMapper
ARMCortexA73CT.RAS	PVBusLogger
ARMCortexA73CT.RAS.mapper	PVBusMapper
ARMCortexA73CT.acp_mapper	PVBusMapper
ARMCortexA73CT.cpu0	ARM_Cortex-A73
ARMCortexA73CT.cpu0.UTLB	TLB
ARMCortexA73CT.cpu0.dtlb	TLB
ARMCortexA73CT.cpu0.l1dcache	PVCache
ARMCortexA73CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA73CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA73CT.cpu0.l1icache	PVCache
ARMCortexA73CT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA73CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA73CT.ext_bus	PVBusLogger
ARMCortexA73CT.ext_bus.mapper	PVBusMapper



Name	Instance type
ARMCortexA73CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA73CT.global_debug_rom	debug_rom
ARMCortexA73CT.l2_cache	PVCache
ARMCortexA73CT.l2_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA73CT.l2_cache.upstream[Z] (where Z = 0-16)	PVBusSlave
ARMCortexA73CT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

Name	Component type
ARMCortexA73CT.AMU	PVBusLogger
ARMCortexA73CT.AMU.mapper	PVBusMapper
ARMCortexA73CT.DAP	PVBusLogger
ARMCortexA73CT.DAP.mapper	PVBusMapper
ARMCortexA73CT.DSU	DSU
ARMCortexA73CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA73CT.MMAP	PVBusLogger
ARMCortexA73CT.MMAP.mapper	PVBusMapper
ARMCortexA73CT.RAS	PVBusLogger
ARMCortexA73CT.RAS.mapper	PVBusMapper
ARMCortexA73CT.acp_mapper	PVBusMapper
ARMCortexA73CT.cpu0	ARM_Cortex-A73
ARMCortexA73CT.cpu0.UTLB	TLB
ARMCortexA73CT.cpu0.l1dcache	PVCache
ARMCortexA73CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA73CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA73CT.cpu0.l1icache	PVCache
ARMCortexA73CT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA73CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA73CT.ext_bus	PVBusLogger
ARMCortexA73CT.ext_bus.mapper	PVBusMapper
ARMCortexA73CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA73CT.l2_cache	PVCache
ARMCortexA73CT.l2_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA73CT.l2_cache.upstream[Z] (where Z = 0-16)	PVBusSlave
ARMCortexA73CT.l2_flusher	AsyncCacheFlushUnit

## Ports for ARMCortexA73CT

Port	Direction	Protocol	Description
aa64naa32	slave	Signal	Register width after reset.
acp_s	slave	PVBus	AXI ACP slave port.

Port	Direction	Protocol	Description
broadcastcachemaint	slave	Signal	ACE defined pins.
broadcastinner	slave	Signal	ACE defined pins.
broadcastouter	slave	Signal	ACE defined pins.
cfgend	slave	Signal	This signal if for EE bit initialisation.
cfgsdisable	slave	Signal	This signal disables write access to some secure Interrupt Controller registers.
cfgte	slave	Signal	This signal provides default exception handling state.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
clrexmonack	master	Signal	Acknowledge handshake signal for the clrexmonreq signal
clrexmonreq	slave	Signal	Signals the clearing of an external global exclusive monitor
clusterid	slave	Value	The port sets the value of the affinity levels 1 and 2; bits [23:16] and [15:8] of the MPIDR.
CNTHPIRQ	master	Signal	Timer signals to SOC.
CNTPNSIRQ	master	Signal	Timer signals to SOC.
CNTPSIRQ	master	Signal	Timer signals to SOC.
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.
CNTVIRQ	master	Signal	Timer signals to SOC.
commirq	master	Signal	Interrupt signal from debug communications channel.
commr	master	Signal	Receive portion of Data Transfer Register full.
commtx	master	Signal	Transmit portion of Data Transfer Register empty.
cp15sdisable	slave	Signal	This signal disables write access to some system control processor registers.
cpuporeset	slave	Signal	CPU power on reset.
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
ctidbgirq	master	Signal	Cross trigger matrix port.
dbgack	master	Signal	External debug interface.
dbgen	slave	Signal	External debug interface.
dbgnopwrdown	master	Signal	No power-down request.
dbgpwrupreq	master	Signal	Processor powerup request.
dev_debug_s	slave	PVBus	External debug interface.
edbgrq	slave	Signal	External debug interface.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
l2flushdone	master	Signal	Flush of L2 memory system complete.

Port	Direction	Protocol	Description
l2flushreq	slave	Signal	Request flush of L2 memory system.
l2reset	slave	Signal	Level2 reset.
memorymapped_debug_s	slave	PVBus	External debug interface.
niden	slave	Signal	External debug interface.
periphbase	slave	Value_64	This port sets the base address of private peripheral region.
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
presetdbg	slave	Signal	Debug reset.
pvbus_m0	master	PVBus	The core will generate bus requests on this port.
rei	slave	Signal	Per core RAM Error Interrupt
reset	slave	Signal	Reset.
romaddr	slave	Value_64	Debug ROM base address.
romaddrv	slave	Signal	Debug ROM base address valid.
rvbaraddr	slave	Value_64	Reset vector base address.
sei	slave	Signal	Per core System Error physical pins.
smpen	master	Signal	This signals AMP or SMP mode for each core.
spiden	slave	Signal	External debug interface.
spniden	slave	Signal	External debug interface.
standbywfe	master	Signal	This signal indicates if a core is in WFE state.
standbywfi	master	Signal	This signal indicates if a core is in WFI state
standbywfil2	master	Signal	This signal indicated all cores and L2 are idles and in low power state.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfirq	slave	Signal	Virtualised FIQ.
vinithi	slave	Signal	This signal controls of the location of the exception vectors at reset.
virq	slave	Signal	Virtualised IRQ.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Per core virtual System Error physical pins.

## Parameters for ARMCortexA73CT

### cpuX.AA64nAA32

Type: bool

Default value: true

**cpuX.CFGEND**Type: `bool`Default value: `false`**cpuX.CFGTE**Type: `bool`Default value: `false`**cpuX.CP15SDISABLE**Type: `bool`Default value: `false`**cpuX.CRYPTODISABLE**Type: `bool`Default value: `false`**cpuX.RVBARADDR**Type: `uint64_t`Default value: `0`**cpuX.VINITHI**Type: `bool`Default value: `false`**cpuX.enable\_trace\_special\_hlt\_imm16**Enable usage of parameter `trace_special_hlt_imm16`.Type: `bool`Default value: `false`**cpuX.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint64_t`Default value: `256`**cpuX.min\_sync\_level**force minimum `syncLevel` (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint32_t`

Default value: 0

**`cpuX.semihosting-A32_HLT`**

Type: `uint32_t`

Default value: 0xF000

**`cpuX.semihosting-A64_HLT`**

Type: `uint32_t`

Default value: 0xF000

**`cpuX.semihosting-ARM_SVC`**

Type: `uint32_t`

Default value: 0x123456

**`cpuX.semihosting-T32_HLT`**

Type: `uint32_t`

Default value: 0x3c

**`cpuX.semihosting-Thumb_SVC`**

Type: `uint32_t`

Default value: 0xAB

**`cpuX.semihosting-cmd_line`**

Type: `string`

Default value: ""

**`cpuX.semihosting-cwd`**

Type: `string`

Default value: ""

**`cpuX.semihosting-enable`**

Type: `bool`

Default value: `true`

**`cpuX.semihosting-heap_base`**

Type: `uint32_t`

Default value: 0x0

**cpuX.semihosting-heap\_limit**

Type: uint32\_t

Default value: 0x0F000000

**cpuX.semihosting-stack\_base**

Type: uint32\_t

Default value: 0x10000000

**cpuX.semihosting-stack\_limit**

Type: uint32\_t

Default value: 0x0F000000

**cpuX.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**cpuX.vfp-enable\_at\_reset**

Type: bool

Default value: false

**BROADCASTCACHEMAINT**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

Type: bool

Default value: true

**BROADCASTINNER**

Enable broadcasting of Inner Shareable transactions. The broadcastinner signal will override this value if used.

Type: bool

Default value: true

**BROADCASTOUTER**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

Type: `bool`

Default value: `true`

**CCSIDR-L1D\_override**

If nonzero, override the value presented in CCSIDR for L1D (this is cosmetic and does not affect cache behaviour).

Type: `uint64_t`

Default value: 0

**CCSIDR-L1I\_override**

If nonzero, override the value presented in CCSIDR for L1I (this is cosmetic and does not affect cache behaviour).

Type: `uint64_t`

Default value: 0

**CCSIDR-L2\_override**

If nonzero, override the value presented in CCSIDR for L2 (this is cosmetic and does not affect cache behaviour).

Type: `uint64_t`

Default value: 0

**CLUSTER\_ID**

Type: `uint32_t`

Default value: 0

**DBGROMADDR**

Type: `uint64_t`

Default value: `0x0`

**DBGROMADDRV**

Type: `bool`

Default value: `true`

**GICDISABLE**

Type: `bool`

Default value: `true`

**NUM\_CORES**

Number of cores in cluster.

Type: `uint8_t`

Default value: `1`

**PERIPHBASE**

Type: `uint64_t`

Default value: `0x13080000`

**cpi\_div**

divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `1`

**cpi\_mul**

multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `1`

**dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit.

Type: `uint64_t`

Default value: `0`

**dcache-maintenance\_latency**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0`



**dcache-miss\_latency**

L1 D-Cache timing annotation latency for miss.

Type: `uint64_t`

Default value: 0

**dcache-read\_access\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (based on the size of `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

**dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

**dcache-size**

L1 D-Cache size in bytes.

Type: `uint32_t`

Default value: 0x8000

**dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

**dcache-state\_modelled**

Type: `bool`

Default value: `false`

**dcache-write\_access\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (based on the size of dcache-write\_bus\_width\_in\_bytes. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0

**dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0

**has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: bool

Default value: false

**has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: bool

Default value: false

**icache-hit\_latency**

L1 I-Cache timing annotation latency for hit.

Type: uint64\_t

Default value: 0

**icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when icache-state\_modelled=true.

Type: uint64\_t

Default value: 0

**icache-miss\_latency**

L1 I-Cache timing annotation latency for miss.

Type: `uint64_t`

Default value: 0

**icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

**icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (based on the size of `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

**icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

**icache-state\_modelled**

Type: `bool`

Default value: `false`

**l2cache-hit\_latency**

L2 Cache timing annotation latency for hit.

Type: `uint64_t`

Default value: 0

**l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

**l2cache-miss\_latency**

L2 Cache timing annotation latency for miss.

Type: `uint64_t`

Default value: 0

**l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access (based on the size of `l2cache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

**l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

**l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: 0x80000

**l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

**l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

**l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access (based on the size of `l2cache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

**l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

**ptw\_latency**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint32_t`

Default value: 0

**tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint32_t`

Default value: 0

## 3.72 ARMCortexA75CT

Defined in `LISA/ARMCortexA75CT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r3p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following parameters were added:

- `has_delayed_dbgreg`
- `has_delayed_sysreg`

### About ARMCortexA75CT

The model supports the following features:

- DynamIQ (DSU) system registers.
- Per-core L2 cache.
- A P-Channel for the cluster and for each core.
- Optional peripheral port.
- L3 cache partition.
- Per-core clock.

The model does not support the following features:

- `BROADCASTCACHEMAINTPOU` pin.
- `COREINSTRRET`, `COREINSTRRUN`, or `nPMBIRQ` signals.
- DynamIQ features that are irrelevant to the programmers' view simulation, for example:
  - Automatic CPU retention mode.
  - Level-3 Cache RAM retention.
- 256-bit wide output transactions.
- ECC and parity schemes are hardware-specific so are not supported.
- Self-test features (MBIST).
- Latency configuration.
- Snoop filtering.
- Debug power management signals `DBGPWRDUP` and `DBGIRSTREQ` are not implemented, but `DBGPWRUPREQ` and `DBGNOPWRDWN` are implemented.
- Cache stashing capability.
- Dual ACE masters.
- This model has a variable number of cores per cluster, specified using the `NUM_CORES` parameter. The per-core parameters are preceded by `cpun.`, where `n` identifies the core (0-3).

- This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (crypto.dll or crypto.so) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

## Iris and MTI instances for ARM Cortex-A75CT

This model has the following Iris instances:

Name	Instance type
ARM Cortex-A75CT	Cluster_ARM_Cortex-A75
ARM Cortex-A75CT.AMU	PVBusLogger
ARM Cortex-A75CT.AMU.mapper	PVBusMapper
ARM Cortex-A75CT.DAP	PVBusLogger
ARM Cortex-A75CT.DAP.mapper	PVBusMapper
ARM Cortex-A75CT.DSU	DSU
ARM Cortex-A75CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARM Cortex-A75CT.DSU.mpam_busslave	PVBusSlave
ARM Cortex-A75CT.DSU.shared_cache	PVCache
ARM Cortex-A75CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARM Cortex-A75CT.DSU.shared_cache.upstream[Y] (where Y = 0-3)	PVBusSlave
ARM Cortex-A75CT.MMAP	PVBusLogger
ARM Cortex-A75CT.MMAP.mapper	PVBusMapper
ARM Cortex-A75CT.RAS	PVBusLogger
ARM Cortex-A75CT.RAS.mapper	PVBusMapper
ARM Cortex-A75CT.cpu0	ARM_Cortex-A75
ARM Cortex-A75CT.cpu0.UTLB	TLB
ARM Cortex-A75CT.cpu0.dtlb	TLB
ARM Cortex-A75CT.cpu0.l1dcache	PVCache
ARM Cortex-A75CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARM Cortex-A75CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARM Cortex-A75CT.cpu0.l1licache	PVCache
ARM Cortex-A75CT.cpu0.l1licache.downstream[0].pvbusmaster	PVBusMaster
ARM Cortex-A75CT.cpu0.l1licache.upstream[0]	PVBusSlave
ARM Cortex-A75CT.cpu0.l2cache	PVCache
ARM Cortex-A75CT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARM Cortex-A75CT.cpu0.l2cache.upstream[U] (where U = 0-1)	PVBusSlave
ARM Cortex-A75CT.ext_bus	PVBusLogger
ARM Cortex-A75CT.ext_bus.mapper	PVBusMapper
ARM Cortex-A75CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARM Cortex-A75CT.global_debug_rom	debug_rom

This model has the following MTI trace components:

Name	Component type
ARMCortexA75CT.AMU	PVBusLogger
ARMCortexA75CT.AMU.mapper	PVBusMapper
ARMCortexA75CT.DAP	PVBusLogger
ARMCortexA75CT.DAP.mapper	PVBusMapper
ARMCortexA75CT.DSU	DSU
ARMCortexA75CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA75CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA75CT.DSU.shared_cache	PVCache
ARMCortexA75CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA75CT.DSU.shared_cache.upstream[Y] (where Y = 0-3)	PVBusSlave
ARMCortexA75CT.MMAP	PVBusLogger
ARMCortexA75CT.MMAP.mapper	PVBusMapper
ARMCortexA75CT.RAS	PVBusLogger
ARMCortexA75CT.RAS.mapper	PVBusMapper
ARMCortexA75CT.cpu0	ARM_Cortex-A75
ARMCortexA75CT.cpu0.UTLB	TLB
ARMCortexA75CT.cpu0.l1dcache	PVCache
ARMCortexA75CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA75CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA75CT.cpu0.l1icache	PVCache
ARMCortexA75CT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA75CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA75CT.cpu0.l2cache	PVCache
ARMCortexA75CT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA75CT.cpu0.l2cache.upstream[U] (where U = 0-1)	PVBusSlave
ARMCortexA75CT.ext_bus	PVBusLogger
ARMCortexA75CT.ext_bus.mapper	PVBusMapper
ARMCortexA75CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

## Ports for ARMCortexA75CT

Port	Direction	Protocol	Description
aa64naa32	slave	Signal	Register width after reset.
acp_s	slave	PVBus	AXI ACP slave port
AENDMP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
ASTARTMP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	slave	Signal	CHI defined pins.
broadcastcachemaint	slave	Signal	ACE defined pins.
broadcastouter	slave	Signal	ACE defined pins.
broadcastpersist	slave	Signal	CHI defined pins.



Port	Direction	Protocol	Description
cfgend	slave	Signal	This signal is for EE bit initialisation.
cfgsdisable	slave	Signal	This signal disables write access to some secure Interrupt Controller registers.
cfgte	slave	Signal	This signal provides default exception handling state.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
clusterid	slave	Value	The port sets the value of the affinity levels 2 and 3; bits [39:32] and [23:16] of the MPIDR.
clusterpmuirq	master	Signal	DynamlQ pmu irq
CNTHPIRQ	master	Signal	Timer signals to SOC.
CNTHVIRQ	master	Signal	Timer signals to SOC.
CNTPNSIRQ	master	Signal	Timer signals to SOC.
CNTPSIRQ	master	Signal	Timer signals to SOC.
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.
CNTVIRQ	master	Signal	Timer signals to SOC.
commirq	master	Signal	Interrupt signal from debug communications channel.
core_clk_in	slave	ClockSignal	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
cp15sdisable	slave	Signal	This signal disables write access to some system control processor registers.
cpuporeset	slave	Signal	Power on reset. Initializes all the processor logic, including debug logic.
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
ctidbgirq	master	Signal	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	slave	Signal	External debug interface.
dbgnopwrdown	master	Signal	No power-down request.
dbgpwrupreq	master	Signal	Debug power up request.
dev_debug_s	slave	PVBus	External debug interface.
edbgrq	slave	Signal	External debug interface.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	slave	PVBus	External debug interface.
niden	slave	Signal	External debug interface.
pchannel_cluster	slave	PChannel	PChannel for cluster.
pchannel_core	slave	PChannel	PChannels for cores
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.

Port	Direction	Protocol	Description
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	master	PVBus	The core will generate bus requests on this port.
pvbus_periph_m	master	PVBus	The core can generate peripheral bus request on this port.
rei	slave	Signal	Individual processor RAM Error Interrupt signal input.
reset	slave	Signal	Raising this signal will put the core into reset mode.
rvbaraddr	slave	Value_64	Reset vector base address.
sci_m	master	SystemCoherencyInterface	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei	slave	Signal	Per core virtual System Error physical pins.
spiden	slave	Signal	External debug interface.
spniden	slave	Signal	External debug interface.
sporeset	slave	Signal	A single cluster-wide power on reset signal for all resettable registers in DynamIQ.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfiq	slave	Signal	Virtual FIQ.
vinithi	slave	Signal	This signal controls of the location of the exception vectors at reset.
virq	slave	Signal	Virtual IRQ.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Per core virtual System Error physical pins.

## Parameters for ARMCortexA75CT

### **cpuX.AA64nAA32**

Type: `bool`

Default value: `true`

### **cpuX.CFGEND**

Type: `bool`

Default value: `false`

### **cpuX.CFGTE**

Type: `bool`

Default value: `false`

**cpuX.CP15SDISABLE**Type: `bool`Default value: `false`**cpuX.CRYPTODISABLE**Type: `bool`Default value: `false`**cpuX.RVBARADDR**Type: `uint64_t`Default value: `0`**cpuX.VINITHI**Type: `bool`Default value: `false`**cpuX.enable\_trace\_special\_hlt\_imm16**Enable usage of parameter `trace_special_hlt_imm16`.Type: `bool`Default value: `false`**cpuX.l2cache-hit\_latency**L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.Type: `uint64_t`Default value: `0x0`**cpuX.l2cache-maintenance\_latency**L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.Type: `uint64_t`Default value: `0x0`**cpuX.l2cache-miss\_latency**L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

### **`cpuX.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**cpuX.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint64\_t

Default value: 256

**cpuX.min\_sync\_level**

force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint32\_t

Default value: 0

**cpuX.semihosting-A32\_HLT**

Type: uint32\_t

Default value: 0xF000

**cpuX.semihosting-A64\_HLT**

Type: uint32\_t

Default value: 0xF000

**cpuX.semihosting-ARM\_SVC**

Type: uint32\_t

Default value: 0x123456

**cpuX.semihosting-T32\_HLT**

Type: uint32\_t

Default value: 0x3c

**cpuX.semihosting-Thumb\_SVC**

Type: uint32\_t

Default value: 0xAB

**cpuX.semihosting-cmd\_line**

Type: string

Default value: ""

**cpuX.semihosting-cwd**

Type: string

Default value: ""

**cpuX.semihosting-enable**

Type: bool

Default value: true

**cpuX.semihosting-heap\_base**

Type: uint32\_t

Default value: 0x0

**cpuX.semihosting-heap\_limit**

Type: uint32\_t

Default value: 0x0F000000

**cpuX.semihosting-stack\_base**

Type: uint32\_t

Default value: 0x10000000

**cpuX.semihosting-stack\_limit**

Type: uint32\_t

Default value: 0x0F000000

**cpuX.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**cpuX.vfp-enable\_at\_reset**

Type: bool

Default value: false

**BROADCASTATOMIC**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: bool

Default value: true

**BROADCASTCACHEMAINT**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

Type: bool

Default value: true

**BROADCASTOUTER**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

Type: bool

Default value: true

**BROADCASTPERSIST**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: bool

Default value: true

**CLUSTER\_ID**

Type: uint32\_t

Default value: 0

**GICDISABLE**

Type: bool

Default value: true

**NUM\_CORES**

Number of cores in cluster.

Type: uint8\_t

Default value: 1

**cpi\_div**

divider for calculating CPI (Cycles Per Instruction).

Type: uint32\_t

Default value: 1

**cpi\_mul**

multiplier for calculating CPI (Cycles Per Instruction).

Type: uint32\_t

Default value: 1

**dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit.

Type: uint64\_t

Default value: 0

**dcache-maintenance\_latency**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0

**dcache-miss\_latency**

L1 D-Cache timing annotation latency for miss.

Type: uint64\_t



Default value: 0

#### **dcache-read\_access\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (based on the size of dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0

#### **dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0

#### **dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0

#### **dcache-state\_modelled**

Type: bool

Default value: false

#### **dcache-write\_access\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (based on the size of dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0

#### **dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0

**force\_zero\_PSTATE\_PAN**

Non-architecture parameter to force PSTATE.PAN to be 0. 0: No effect, 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

**has\_acp**

If true, Accelerator Coherency Port is configured.

Type: `bool`

Default value: `false`

**has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

**has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**has\_peripheral\_port**

If true, additional AXI peripheral port is configured.

Type: `bool`

Default value: `false`

**icache-hit\_latency**

L1 I-Cache timing annotation latency for hit.

Type: `uint64_t`

Default value: 0

**icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

**icache-miss\_latency**

L1 I-Cache timing annotation latency for miss.

Type: `uint64_t`

Default value: 0

**icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

**icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (based on the size of `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

**icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: 0

**icache-state\_modelled**

Type: `bool`

Default value: `false`

**l3cache-hit\_latency**

L3 Cache timing annotation latency for hit.

Type: uint64\_t

Default value: 0

**l3cache-maintenance\_latency**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0

**l3cache-miss\_latency**

L3 Cache timing annotation latency for miss.

Type: uint64\_t

Default value: 0

**l3cache-read\_access\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per access (based on the size of l3cache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-read\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0

**l3cache-read\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0

**l3cache-size**

L3 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**l3cache-snoop\_data\_transfer\_latency**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0

**l3cache-snoop\_issue\_latency**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0

**l3cache-write\_access\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per access (based on the size of l3cache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0

**l3cache-write\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0

**periph\_address\_end**

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

Type: uint64\_t

Default value: 0x0

**periph\_address\_start**

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

Type: uint64\_t

Default value: 0x0

**ptw\_latency**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint32\_t

Default value: 0

**tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint32\_t

Default value: 0

## 3.73 ARMCortexA76AECT

Defined in `LISA/ARMCortexA76AECT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following parameters were added:

- `has_delayed_dbgreg`
- `has_delayed_sysreg`

### About ARMCortexA76AECT

The model supports the following features:

- DynamIQ (DSU) system registers.
- Per-core L2 cache.
- A P-Channel for the cluster and for each core.
- Optional peripheral port.
- L3 cache partition.
- Per-core clock.
- `BROADCASTPERSIST` pin.

The model does not support the following features:

- `BROADCASTCACHEMAINTPOU` pin.

- `COREINSTRET`, `COREINSTRUN`, or `nPMBIRQ` signals.
- DynamIQ features that are irrelevant to the programmers' view simulation, for example:
  - Automatic CPU retention mode.
  - Level-3 Cache RAM retention.
- 256-bit wide output transactions.
- ECC and parity schemes are hardware-specific so are not supported.
- Self-test features (MBIST).
- Latency configuration.
- Snoop filtering.
- Debug power management signals `DBGPWRDUP` and `DBGIRSTREQ` are not implemented, but `DBGPWRUPREQ` and `DBGNOPWRDWN` are implemented.
- Cache stashing capability.
- Split/Lock is supported but with the limitations described in the AE-specific features implemented section.

This model has a variable number of cores per cluster, specified using the `NUM_CORES` parameter.

The per-core parameters are preceded by `cpun.`, where `n` identifies the core (0-3).

This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (`crypto.dll` or `crypto.so`) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

### AE-specific features implemented

The only AE-specific features implemented in the model are Split-Lock and Lock-Step, with the following limitations:

- The implementation of Split-Lock and Lock-Step in the model is simply a halving of the number of cores available.
- Lock-step is implemented by a parameter that can be true or false for the entire execution only, rather than by a signal that can be enabled during execution.

As the model does not implement true Split-Lock functionality, the ports and parameters are contiguous. For example, for a 4-core CPU with Split-Lock enabled, `cpu0` and `cpu1` identify the available cores and associated ports, not `cpu0` and `cpu2`.

- Hybrid mode is not modeled in the DSU.

### Iris and MTI instances for ARM CortexA76AECT

This model has the following Iris instances:

Name	Instance type
ARM CortexA76AECT	Cluster_ARM_Cortex-A76AE
ARM CortexA76AECT.AMU	PVBusLogger

Name	Instance type
ARMCortexA76AECT.AMU.mapper	PVBusMapper
ARMCortexA76AECT.DAP	PVBusLogger
ARMCortexA76AECT.DAP.mapper	PVBusMapper
ARMCortexA76AECT.DSU	DSU
ARMCortexA76AECT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA76AECT.DSU.mpam_busslave	PVBusSlave
ARMCortexA76AECT.DSU.shared_cache	PVCache
ARMCortexA76AECT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA76AECT.DSU.shared_cache.upstream[Y] (where Y = 0–5)	PVBusSlave
ARMCortexA76AECT.MMAP	PVBusLogger
ARMCortexA76AECT.MMAP.mapper	PVBusMapper
ARMCortexA76AECT.RAS	PVBusLogger
ARMCortexA76AECT.RAS.mapper	PVBusMapper
ARMCortexA76AECT.cpuY (where Y = 0–1)	ARM_Cortex-A76AE
ARMCortexA76AECT.cpuY.UTLB (where Y = 0–1)	TLB
ARMCortexA76AECT.cpuY.debug_rom (where Y = 0–1)	debug_rom
ARMCortexA76AECT.cpuY.dtlb (where Y = 0–1)	TLB
ARMCortexA76AECT.cpuY.l1dcache (where Y = 0–1)	PVCache
ARMCortexA76AECT.cpuY.l1dcache.downstream[0].pvbusmaster (where Y = 0–1)	PVBusMaster
ARMCortexA76AECT.cpuY.l1dcache.upstream[0] (where Y = 0–1)	PVBusSlave
ARMCortexA76AECT.cpuY.l1licache (where Y = 0–1)	PVCache
ARMCortexA76AECT.cpuY.l1licache.downstream[0].pvbusmaster (where Y = 0–1)	PVBusMaster
ARMCortexA76AECT.cpuY.l1licache.upstream[0] (where Y = 0–1)	PVBusSlave
ARMCortexA76AECT.cpuY.l2cache (where Y = 0–1)	PVCache
ARMCortexA76AECT.cpuY.l2cache.downstream[0].pvbusmaster (where Y = 0–1)	PVBusMaster
ARMCortexA76AECT.cpuY.l2cache.upstream[U] (where Y = 0–1; U = 0–1)	PVBusSlave
ARMCortexA76AECT.ext_bus	PVBusLogger
ARMCortexA76AECT.ext_bus.mapper	PVBusMapper
ARMCortexA76AECT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA76AECT.global_debug_rom	debug_rom
ARMCortexA76AECT.secondary_debug_rom	debug_rom

This model has the following MTI trace components:

Name	Component type
ARMCortexA76AECT.AMU	PVBusLogger
ARMCortexA76AECT.AMU.mapper	PVBusMapper
ARMCortexA76AECT.DAP	PVBusLogger
ARMCortexA76AECT.DAP.mapper	PVBusMapper
ARMCortexA76AECT.DSU	DSU
ARMCortexA76AECT.DSU.l3_flusher	AsyncCacheFlushUnit



Name	Component type
ARMCortexA76AECT.DSU.mpam_busslave	PVBusSlave
ARMCortexA76AECT.DSU.shared_cache	PVCache
ARMCortexA76AECT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA76AECT.DSU.shared_cache.upstream[Y] (where Y = 0-5)	PVBusSlave
ARMCortexA76AECT.MMAP	PVBusLogger
ARMCortexA76AECT.MMAP.mapper	PVBusMapper
ARMCortexA76AECT.RAS	PVBusLogger
ARMCortexA76AECT.RAS.mapper	PVBusMapper
ARMCortexA76AECT.cpuY (where Y = 0-1)	ARM_Cortex-A76AE
ARMCortexA76AECT.cpuY.UTLB (where Y = 0-1)	TLB
ARMCortexA76AECT.cpuY.l1dcache (where Y = 0-1)	PVCache
ARMCortexA76AECT.cpuY.l1dcache.downstream[0].pvbusmaster (where Y = 0-1)	PVBusMaster
ARMCortexA76AECT.cpuY.l1dcache.upstream[0] (where Y = 0-1)	PVBusSlave
ARMCortexA76AECT.cpuY.l1licache (where Y = 0-1)	PVCache
ARMCortexA76AECT.cpuY.l1licache.downstream[0].pvbusmaster (where Y = 0-1)	PVBusMaster
ARMCortexA76AECT.cpuY.l1licache.upstream[0] (where Y = 0-1)	PVBusSlave
ARMCortexA76AECT.cpuY.l2cache (where Y = 0-1)	PVCache
ARMCortexA76AECT.cpuY.l2cache.downstream[0].pvbusmaster (where Y = 0-1)	PVBusMaster
ARMCortexA76AECT.cpuY.l2cache.upstream[U] (where Y = 0-1; U = 0-1)	PVBusSlave
ARMCortexA76AECT.ext_bus	PVBusLogger
ARMCortexA76AECT.ext_bus.mapper	PVBusMapper
ARMCortexA76AECT.gic_cpuif_decoder_cluster	GIcV3CPUInterfaceDecoder

## Ports for ARMCortexA76AECT

Port	Direction	Protocol	Description
acp_s	slave	PVBus	AXI ACP slave port.
AENDMP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
ASTARTMP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	slave	Signal	CHI defined pins.
broadcastcachemaint	slave	Signal	ACE defined pins.
broadcastouter	slave	Signal	ACE defined pins.
broadcastpersist	slave	Signal	CHI defined pins.
cfgend	slave	Signal	This signal if for EE bit initialisation.
cfgte	slave	Signal	This signal provides default exception handling state.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.

Port	Direction	Protocol	Description
clusterid	slave	Value	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set ManagerID64 of the core.
clusterpmuirq	master	Signal	DynamiQ pmu irq
CNTHPIRQ	master	Signal	Timer signals to SOC.
CNTHVIRQ	master	Signal	Timer signals to SOC.
CNTPNSIRQ	master	Signal	Timer signals to SOC.
CNTPSIRQ	master	Signal	Timer signals to SOC.
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.
CNTVIRQ	master	Signal	Timer signals to SOC.
commirq	master	Signal	Interrupt signal from debug communications channel.
core_clk_in	slave	ClockSignal	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
cpuporeset	slave	Signal	Power on reset. Initializes all the processor logic, including debug logic.
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
ctidbgirq	master	Signal	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	slave	Signal	External debug interface.
dbgnopwrdown	master	Signal	These signals relate to core power down.
dbgpwrupreq	master	Signal	Debug power up request.
dev_debug_s	slave	PVBus	External debug interface.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	slave	PVBus	External debug interface.
niden	slave	Signal	External debug interface.
pchannel_cluster	slave	PChannel	PChannel for cluster.
pchannel_core	slave	PChannel	PChannels for cores
pmbirq	master	Signal	Interrupt signal from statistical profiling unit.
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	master	PVBus	The core will generate bus requests on this port.

Port	Direction	Protocol	Description
pvbus_periph_m	master	PVBus	The core can generate peripheral bus request on this port.
rei	slave	Signal	Per core RAM Error Interrupt.
reset	slave	Signal	Raising this signal will put the core into reset mode.
rvbaraddr	slave	Value_64	Reset vector base address.
sci_m	master	SystemCoherencyInterface	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei	slave	Signal	Per core virtual System Error physical pins.
spiden	slave	Signal	External debug interface.
spniden	slave	Signal	External debug interface.
sporeset	slave	Signal	A single cluster-wide power on reset signal for all resettable registers in DynamIQ.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfiq	slave	Signal	Virtualised FIQ.
vinithi	slave	Signal	This signal controls of the location of the exception vectors at reset.
virq	slave	Signal	Virtualised IRQ.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Per core virtual System Error physical pins.

## Parameters for ARM Cortex A76 AECT

### cpuX.CFGEND

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### cpuX.CFGTE

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### cpuX.CRYPTODISABLE

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`cpuX.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.VINITHI`**

Reset value of SCTL.R.V.

Type: `bool`

Default value: `false`

### **`cpuX.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`cpuX.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**cpuX.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**cpuX.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

### **`cpuX.min_sync_level`**

Force minimum `syncLevel` (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

### **`cpuX.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`cpuX.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`cpuX.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**cpuX.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**cpuX.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**cpuX.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**cpuX.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**cpuX.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**cpuX.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: 0x0

**cpuX.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: 0xf000000

**cpuX.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**cpuX.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**cpuX.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**cpuX.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**BROADCASTATOMIC**

Enable broadcasting of atomic operation. The `broadcastatomic` signal will override this value if used.

Type: `bool`

Default value: `true`

**BROADCASTACHEMAINT**

Enable broadcasting of cache maintenance operations to downstream caches. The `broadcastcachemaint` signal will override this value if used.

Type: `bool`

Default value: `true`



**BROADCASTOUTER**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

Type: `bool`

Default value: `true`

**BROADCASTPERSIST**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: `bool`

Default value: `true`

**CLUSTER\_ID**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the ManagerID64 of the core.

Type: `uint32_t`

Default value: `0x0`

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. This value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required.

Type: `uint8_t`

Default value: `1`

**GICDISABLE**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`

Default value: `true`

**NUM\_CORES**

Number of cores per cluster.

Type: `uint8_t`

Default value: 2

**cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: 0x1

**cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: 0x1

**dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**dcache-maintenance\_latency**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**dcache-miss\_latency**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**dcache-prefetch\_enabled**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

Type: `bool`

Default value: `false`

### **`dcache-read_access_latency`**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`dcache-read_latency`**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`dcache-snoop_data_transfer_latency`**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`dcache-state_modelled`**

Set whether D-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **`dcache-write_access_latency`**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **default\_opmode**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

Type: uint8\_t

Default value: 4

### **diagnostics**

Enable DynamIQ diagnostic messages.

Type: bool

Default value: false

### **enable\_lock\_step**

Whether the core is configured in Dual Core Lock Step mode (FEAT\_DCLS).

Type: bool

Default value: false

### **enable\_simulation\_performance\_optimizations**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

Type: bool

Default value: true

### **ext\_abort\_device\_read\_is\_sync**

Synchronous reporting of device-nGnRE read external aborts.

Type: bool

Default value: `false`

**ext\_abort\_device\_write\_is\_sync**

Synchronous reporting of device-nGnRE write external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_so\_read\_is\_sync**

Synchronous reporting of device-nGnRnE read external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_so\_write\_is\_sync**

Synchronous reporting of device-nGnRnE write external aborts.

Type: `bool`

Default value: `false`

**force\_zero\_PSTATE\_PAN**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

**has\_acp**

If true, Accelerator Coherency Port is configured.

Type: `bool`

Default value: `false`

**has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

### **has\_dot\_product**

Possible values of this parameter are:- 1, feature not implemented.- 2, feature is implemented.

Type: `uint8_t`

Default value: 2

### **has\_peripheral\_port**

If true, additional AXI peripheral port is configured.

Type: `bool`

Default value: `false`

### **has\_statistical\_profiling**

Whether Statistical Based Profiling is implemented (FEAT\_SPE).

Type: `bool`

Default value: `true`

### **icache-hit\_latency**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

Type: `bool`

Default value: `false`

**icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when icache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**icache-state\_modelled**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

**invalidate\_code\_cache\_on\_icache\_cmo**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: `uint8_t`

Default value: `0`

**l3cache-hit\_latency**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

### **l3cache-maintenance\_latency**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **l3cache-miss\_latency**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **l3cache-read\_access\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size `l3cache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **l3cache-read\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l3cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **l3cache-size**

L3 Cache size in bytes.

Type: `uint32_t`

Default value: `0x100000`



**l3cache-snoop\_data\_transfer\_latency**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-snoop\_issue\_latency**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-write\_access\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size `l3cache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-write\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l3cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**pchannel\_treat\_simreset\_as\_poreset**

Register core as ON state to cluster with simulation reset.

Type: `bool`

Default value: `false`

**periph\_address\_end**

End address for peripheral port address range exclusive (corresponds to AENDMP input signal).

Type: `uint64_t`

Default value: `0x0`

**periph\_address\_start**

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

Type: uint64\_t

Default value: 0x0

**ptw\_latency**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

**reported\_patch\_level**

Purely cosmetic patch level value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: int8\_t

Default value: -1

**reported\_revision\_number**

Purely cosmetic revision number value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: int8\_t

Default value: -1

**tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

**tlbi\_stall\_enabled**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: bool

Default value: false

**treat-dcache-cmos-to-pou-as-nop**

Whether dcache invalidation to the point of unification is required for instruction to data coherence. 0 - Invalidate ops required, 1 - Invalidate ops not required and cannot generate faults, 2 - Invalidate ops not required but can generate faults.

Type: `uint8_t`

Default value: 0

**walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

## 3.74 ARMCortexA76CT

Defined in `LISA/ARMCortexA76CT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

**Changes in 11.31.15**

The following parameters were added:

- `has_delayed_dbgreg`
- `has_delayed_sysreg`

**About ARMCortexA76CT**

The model supports the following features:

- DynamIQ (DSU) system registers.
- Per-core L2 cache.
- A P-Channel for the cluster and for each core.
- Optional peripheral port.
- L3 cache partition.
- Per-core clock.
- `BROADCASTPERSIST` pin.

The model does not support the following features:

- BROADCASTCACHEMAINTPOU pin.
- COREINSTRRET, COREINSTRRUN, or nPMBIRQ signals.
- DynamIQ features that are irrelevant to the programmers' view simulation, for example:
  - Automatic CPU retention mode.
  - Level-3 Cache RAM retention.
- 256-bit wide output transactions.
- ECC and parity schemes are hardware-specific so are not supported.
- Self-test features (MBIST).
- Latency configuration.
- Snoop filtering.
- Debug power management signals DBGWRUP and DBGRSTREQ are not implemented, but DBGWRUPREQ and DBGNOFWRDWN are implemented.
- Cache stashing capability.

This model has a variable number of cores per cluster, specified using the `NUM_CORES` parameter.

The per-core parameters are preceded by `cpun.`, where `n` identifies the core (0-3).

This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (`crypto.dll` or `crypto.so`) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

## Iris and MTI instances for ARM Cortex-A76CT

This model has the following Iris instances:

Name	Instance type
ARM Cortex-A76CT	Cluster_ARM_Cortex-A76
ARM Cortex-A76CT.AMU	PVBusLogger
ARM Cortex-A76CT.AMU.mapper	PVBusMapper
ARM Cortex-A76CT.DAP	PVBusLogger
ARM Cortex-A76CT.DAP.mapper	PVBusMapper
ARM Cortex-A76CT.DSU	DSU
ARM Cortex-A76CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARM Cortex-A76CT.DSU.mpam_busslave	PVBusSlave
ARM Cortex-A76CT.DSU.shared_cache	PVCache
ARM Cortex-A76CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARM Cortex-A76CT.DSU.shared_cache.upstream[Y] (where Y = 0-3)	PVBusSlave
ARM Cortex-A76CT.MMAP	PVBusLogger
ARM Cortex-A76CT.MMAP.mapper	PVBusMapper
ARM Cortex-A76CT.RAS	PVBusLogger

Name	Instance type
ARMCortexA76CT.RAS.mapper	PVBusMapper
ARMCortexA76CT.cpu0	ARM_Cortex-A76
ARMCortexA76CT.cpu0.UTLB	TLB
ARMCortexA76CT.cpu0.debug_rom	debug_rom
ARMCortexA76CT.cpu0.dtlb	TLB
ARMCortexA76CT.cpu0.l1dcache	PVCache
ARMCortexA76CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA76CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA76CT.cpu0.l1icache	PVCache
ARMCortexA76CT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA76CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA76CT.cpu0.l2cache	PVCache
ARMCortexA76CT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA76CT.cpu0.l2cache.upstream[U] (where $U = 0-1$ )	PVBusSlave
ARMCortexA76CT.ext_bus	PVBusLogger
ARMCortexA76CT.ext_bus.mapper	PVBusMapper
ARMCortexA76CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA76CT.global_debug_rom	debug_rom
ARMCortexA76CT.secondary_debug_rom	debug_rom

This model has the following MTI trace components:

Name	Component type
ARMCortexA76CT.AMU	PVBusLogger
ARMCortexA76CT.AMU.mapper	PVBusMapper
ARMCortexA76CT.DAP	PVBusLogger
ARMCortexA76CT.DAP.mapper	PVBusMapper
ARMCortexA76CT.DSU	DSU
ARMCortexA76CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA76CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA76CT.DSU.shared_cache	PVCache
ARMCortexA76CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA76CT.DSU.shared_cache.upstream[Y] (where $Y = 0-3$ )	PVBusSlave
ARMCortexA76CT.MMAP	PVBusLogger
ARMCortexA76CT.MMAP.mapper	PVBusMapper
ARMCortexA76CT.RAS	PVBusLogger
ARMCortexA76CT.RAS.mapper	PVBusMapper
ARMCortexA76CT.cpu0	ARM_Cortex-A76
ARMCortexA76CT.cpu0.UTLB	TLB
ARMCortexA76CT.cpu0.l1dcache	PVCache
ARMCortexA76CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster

Name	Component type
ARMCortexA76CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA76CT.cpu0.l1icache	PVCache
ARMCortexA76CT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA76CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA76CT.cpu0.l2cache	PVCache
ARMCortexA76CT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA76CT.cpu0.l2cache.upstream[U] (where U = 0-1)	PVBusSlave
ARMCortexA76CT.ext_bus	PVBusLogger
ARMCortexA76CT.ext_bus.mapper	PVBusMapper
ARMCortexA76CT.gic_cpuif_decoder_cluster	GLCv3CPUInterfaceDecoder

## Ports for ARMCortexA76CT

Port	Direction	Protocol	Description
acp_s	slave	PVBus	AXI ACP slave port.
AENDMP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
ASTARTMP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	slave	Signal	CHI defined pins.
broadcastcachemaint	slave	Signal	ACE defined pins.
broadcastouter	slave	Signal	ACE defined pins.
broadcastpersist	slave	Signal	CHI defined pins.
cfgend	slave	Signal	This signal if for EE bit initialisation.
cfgsdisable	slave	Signal	This signal disables write access to some secure Interrupt Controller registers.
cfgte	slave	Signal	This signal provides default exception handling state.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
clusterid	slave	Value	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set ManagerID64 of the core.
clusterpmuirq	master	Signal	DynamiQ pmu irq
CNTHPIRQ	master	Signal	Timer signals to SOC.
CNTHVIRQ	master	Signal	Timer signals to SOC.
CNTPNSIRQ	master	Signal	Timer signals to SOC.
CNTPSIRQ	master	Signal	Timer signals to SOC.

Port	Direction	Protocol	Description
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.
CNTVIRQ	master	Signal	Timer signals to SOC.
commirq	master	Signal	Interrupt signal from debug communications channel.
core_clk_in	slave	ClockSignal	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
cp15sdisable	slave	Signal	This signal disables write access to some system control processor registers.
cpuporeset	slave	Signal	Power on reset. Initializes all the processor logic, including debug logic.
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
ctidbgirq	master	Signal	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	slave	Signal	External debug interface.
dbgnopwrdown	master	Signal	These signals relate to core power down.
dbgpwrupreq	master	Signal	Debug power up request.
dev_debug_s	slave	PVBus	External debug interface.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	slave	PVBus	External debug interface.
niden	slave	Signal	External debug interface.
pchannel_cluster	slave	PChannel	PChannel for cluster.
pchannel_core	slave	PChannel	PChannels for cores
pmbirq	master	Signal	Interrupt signal from statistical profiling unit.
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	master	PVBus	The core will generate bus requests on this port.
pvbus_periph_m	master	PVBus	The core can generate peripheral bus request on this port.
rei	slave	Signal	Per core RAM Error Interrupt.
reset	slave	Signal	Raising this signal will put the core into reset mode.
rvbaraddr	slave	Value_64	Reset vector base address.
sci_m	master	SystemCoherencyInterface	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei	slave	Signal	Per core virtual System Error physical pins.
spiden	slave	Signal	External debug interface.
spniden	slave	Signal	External debug interface.

Port	Direction	Protocol	Description
sporeset	slave	Signal	A single cluster-wide power on reset signal for all resettable registers in DynamIQ.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfirq	slave	Signal	Virtualised FIQ.
vinithi	slave	Signal	This signal controls of the location of the exception vectors at reset.
virq	slave	Signal	Virtualised IRQ.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Per core virtual System Error physical pins.

## Parameters for ARM Cortex A76CT

### **cpuX.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **cpuX.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **cpuX.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **cpuX.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`



**cpuX.VINITHI**

Reset value of SCTLR.V.

Type: `bool`

Default value: `false`

**cpuX.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**cpuX.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**cpuX.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**cpuX.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**cpuX.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**cpuX.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**cpuX.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**cpuX.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**cpuX.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**cpuX.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**cpuX.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**cpuX.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**cpuX.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

### **`cpuX.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **`cpuX.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **`cpuX.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`cpuX.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`cpuX.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`cpuX.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

#### **`cpuX.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`cpuX.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **BROADCASTATOMIC**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`

Default value: `true`

#### **BROADCASTCACHEMAINT**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

Type: `bool`

Default value: `true`

#### **BROADCASTOUTER**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

Type: `bool`

Default value: `true`

#### **BROADCASTPERSIST**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: `bool`

Default value: `true`

### **CLUSTER\_ID**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the ManagerID64 of the core.

Type: `uint32_t`

Default value: `0x0`

### **CMO\_broadcast\_when\_cache\_state\_modelling\_disabled**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. This value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required.

Type: `uint8_t`

Default value: `1`

### **GICDISABLE**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`

Default value: `true`

### **NUM\_CORES**

Number of cores per cluster.

Type: `uint8_t`

Default value: `1`

### **cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: 0x1

**cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: uint32\_t

Default value: 0x1

**dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**dcache-maintenance\_latency**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**dcache-miss\_latency**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**dcache-prefetch\_enabled**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

Type: bool

Default value: false

**dcache-read\_access\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **dcache-state\_modelled**

Set whether D-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **dcache-write\_access\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0



**default\_opmode**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

Type: `uint8_t`

Default value: 4

**diagnostics**

Enable DynamIQ diagnostic messages.

Type: `bool`

Default value: `false`

**enable\_simulation\_performance\_optimizations**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of `stage12_tlb_size` parameter to 1024).

Type: `bool`

Default value: `true`

**ext\_abort\_device\_read\_is\_sync**

Synchronous reporting of device-nGnRE read external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_device\_write\_is\_sync**

Synchronous reporting of device-nGnRE write external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_so\_read\_is\_sync**

Synchronous reporting of device-nGnRnE read external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_so\_write\_is\_sync**

Synchronous reporting of device-nGnRnE write external aborts.

Type: `bool`

Default value: `false`

**force\_zero\_PSTATE\_PAN**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

**has\_acp**

If true, Accelerator Coherency Port is configured.

Type: `bool`

Default value: `false`

**has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**has\_peripheral\_port**

If true, additional AXI peripheral port is configured.

Type: `bool`

Default value: `false`

**has\_statistical\_profiling**

Whether Statistical Based Profiling is implemented (FEAT\_SPE).

Type: `bool`

Default value: `true`

### **`icache-hit_latency`**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`icache-maintenance_latency`**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`icache-miss_latency`**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`icache-prefetch_enabled`**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

### **`icache-read_access_latency`**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`icache-read_latency`**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to

the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`icache-state_modelled`**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **`invalidate_code_cache_on_icache_cmo`**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: `uint8_t`

Default value: `0`

### **`l3cache-hit_latency`**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-maintenance_latency`**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-miss_latency`**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-read\_access\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size l3cache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**l3cache-read\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. l3cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**l3cache-size**

L3 Cache size in bytes.

Type: uint32\_t

Default value: 0x100000

**l3cache-snoop\_data\_transfer\_latency**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**l3cache-snoop\_issue\_latency**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**l3cache-write\_access\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size l3cache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used

instead of per-byte even if l3cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **l3cache-write\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. l3cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **pchannel\_treat\_simreset\_as\_poreset**

Register core as ON state to cluster with simulation reset.

Type: bool

Default value: false

### **periph\_address\_end**

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

Type: uint64\_t

Default value: 0x0

### **periph\_address\_start**

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

Type: uint64\_t

Default value: 0x0

### **ptw\_latency**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

### **tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

### **tlbi\_stall\_enabled**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: bool

Default value: false

### **treat-dcache-cmos-to-pou-as-nop**

Whether dcache invalidation to the point of unification is required for instruction to data coherence. 0 - Invalidate ops required, 1 - Invalidate ops not required and cannot generate faults, 2 - Invalidate ops not required but can generate faults.

Type: uint8\_t

Default value: 0

### **walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

## 3.75 ARMCortexA77CT

Defined in LISA/ARMCortexA77CT.lisa.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following parameters were added:

- has\_delayed\_dbgreg
- has\_delayed\_sysreg

### About ARMCortexA77CT

The model supports the following features:

- DynamIQ (DSU) system registers.

- Per-core L2 cache.
- A P-Channel for the cluster and for each core.
- Optional peripheral port.
- L3 cache partition.
- Per-core clock.
- `BROADCASTPERSIST` pin.

The model does not support the following features:

- `BROADCASTCACHEMAINTPOU` pin.
- `COREINSTRRET`, `COREINSTRRUN`, or `nPMBIRQ` signals.
- DynamIQ features that are irrelevant to the programmers' view simulation, for example:
  - Automatic CPU retention mode.
  - Level-3 Cache RAM retention.
- 256-bit wide output transactions.
- ECC and parity schemes are hardware-specific so are not supported.
- Self-test features (MBIST).
- Latency configuration.
- Snoop filtering.
- Debug power management signals `DBGPWRDUP` and `DBGIRSTREQ` are not implemented, but `DBGPWRUPREQ` and `DBGNOPWRDWN` are implemented.
- Cache stashing capability.

This model has a variable number of cores per cluster, specified using the `NUM_CORES` parameter. The per-core parameters are preceded by `cpun.`, where `n` identifies the core (0-3).

This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (`crypto.dll` or `crypto.so`) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

## Iris and MTI instances for ARM Cortex A77CT

This model has the following Iris instances:

Name	Instance type
ARM Cortex A77CT	Cluster_ARM_Cortex-A77
ARM Cortex A77CT.AMU	PVBusLogger
ARM Cortex A77CT.AMU.mapper	PVBusMapper
ARM Cortex A77CT.DAP	PVBusLogger
ARM Cortex A77CT.DAP.mapper	PVBusMapper
ARM Cortex A77CT.DSU	DSU
ARM Cortex A77CT.DSU.l3_flusher	AsyncCacheFlushUnit



Name	Instance type
ARMCortexA77CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA77CT.DSU.shared_cache	PVCache
ARMCortexA77CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA77CT.DSU.shared_cache.upstream[Y] (where Y = 0–3)	PVBusSlave
ARMCortexA77CT.MMAP	PVBusLogger
ARMCortexA77CT.MMAP.mapper	PVBusMapper
ARMCortexA77CT.RAS	PVBusLogger
ARMCortexA77CT.RAS.mapper	PVBusMapper
ARMCortexA77CT.cpu0	ARM_Cortex-A77
ARMCortexA77CT.cpu0.UTLB	TLB
ARMCortexA77CT.cpu0.debug_rom	debug_rom
ARMCortexA77CT.cpu0.dtlb	TLB
ARMCortexA77CT.cpu0.l1dcache	PVCache
ARMCortexA77CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA77CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA77CT.cpu0.l1icache	PVCache
ARMCortexA77CT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA77CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA77CT.cpu0.l2cache	PVCache
ARMCortexA77CT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA77CT.cpu0.l2cache.upstream[U] (where U = 0–1)	PVBusSlave
ARMCortexA77CT.ext_bus	PVBusLogger
ARMCortexA77CT.ext_bus.mapper	PVBusMapper
ARMCortexA77CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA77CT.global_debug_rom	debug_rom
ARMCortexA77CT.secondary_debug_rom	debug_rom

This model has the following MTI trace components:

Name	Component type
ARMCortexA77CT.AMU	PVBusLogger
ARMCortexA77CT.AMU.mapper	PVBusMapper
ARMCortexA77CT.DAP	PVBusLogger
ARMCortexA77CT.DAP.mapper	PVBusMapper
ARMCortexA77CT.DSU	DSU
ARMCortexA77CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA77CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA77CT.DSU.shared_cache	PVCache
ARMCortexA77CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA77CT.DSU.shared_cache.upstream[Y] (where Y = 0–3)	PVBusSlave
ARMCortexA77CT.MMAP	PVBusLogger

Name	Component type
ARMCortexA77CT.MMAP.mapper	PVBusMapper
ARMCortexA77CT.RAS	PVBusLogger
ARMCortexA77CT.RAS.mapper	PVBusMapper
ARMCortexA77CT.cpu0	ARM_Cortex-A77
ARMCortexA77CT.cpu0.UTLB	TLB
ARMCortexA77CT.cpu0.l1dcache	PVCache
ARMCortexA77CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA77CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA77CT.cpu0.l1icache	PVCache
ARMCortexA77CT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA77CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA77CT.cpu0.l2cache	PVCache
ARMCortexA77CT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA77CT.cpu0.l2cache.upstream[U] (where $U = 0-1$ )	PVBusSlave
ARMCortexA77CT.ext_bus	PVBusLogger
ARMCortexA77CT.ext_bus.mapper	PVBusMapper
ARMCortexA77CT.gic_cpuif_decoder_cluster	GLCv3CPUInterfaceDecoder

### Ports for ARMCortexA77CT

Port	Direction	Protocol	Description
acp_s	slave	PVBus	AXI ACP slave port.
AENDMP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
ASTARTMP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	slave	Signal	CHI defined pins.
broadcastcachemaint	slave	Signal	ACE defined pins.
broadcastouter	slave	Signal	ACE defined pins.
broadcastpersist	slave	Signal	CHI defined pins.
cfgend	slave	Signal	This signal if for EE bit initialisation.
cfgsdisable	slave	Signal	This signal disables write access to some secure Interrupt Controller registers.
cfgte	slave	Signal	This signal provides default exception handling state.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.

Port	Direction	Protocol	Description
clusterid	slave	Value	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set ManagerID64 of the core.
clusterpmuirq	master	Signal	DynamiQ pmu irq
CNTHPIRQ	master	Signal	Timer signals to SOC.
CNTHVIRQ	master	Signal	Timer signals to SOC.
CNTPNSIRQ	master	Signal	Timer signals to SOC.
CNTPSIRQ	master	Signal	Timer signals to SOC.
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.
CNTVIRQ	master	Signal	Timer signals to SOC.
commirq	master	Signal	Interrupt signal from debug communications channel.
core_clk_in	slave	ClockSignal	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
cpuporeset	slave	Signal	CPU power on reset. Initializes all the processor logic, including debug logic.
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
ctidbgirq	master	Signal	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	slave	Signal	External debug interface.
dbgnopwrdown	master	Signal	No power-down request.
dbgpwrupreq	master	Signal	Debug power up request.
dev_debug_s	slave	PVBus	External debug interface.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	slave	PVBus	External debug interface.
niden	slave	Signal	External debug interface.
pchannel_cluster	slave	PChannel	PChannel for cluster.
pchannel_core	slave	PChannel	PChannels for cores
pmbirq	master	Signal	Interrupt signal from statistical profiling unit.
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	master	PVBus	The core will generate bus requests on this port.

Port	Direction	Protocol	Description
pvbus_periph_m	master	PVBus	The core can generate peripheral bus request on this port.
rei	slave	Signal	Per core RAM Error Interrupt.
reset	slave	Signal	Raising this signal will put the core into reset mode.
rvbaraddr	slave	Value_64	Reset vector base address.
sci_m	master	SystemCoherencyInterface	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei	slave	Signal	Per core System Error physical pins.
spiden	slave	Signal	External debug interface.
spniden	slave	Signal	External debug interface.
sporeset	slave	Signal	A single cluster-wide power on reset signal for all resettable registers in DynamIQ.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfiq	slave	Signal	Virtualised FIQ.
vinithi	slave	Signal	This signal controls of the location of the exception vectors at reset.
virq	slave	Signal	Virtualised IRQ.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Per core virtual System Error physical pins.

## Parameters for ARM Cortex A77CT

### cpuX.CFGEND

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### cpuX.CFGTE

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### cpuX.CRYPTODISABLE

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **`cpuX.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.VINITHI`**

Reset value of SCTL.R.V.

Type: `bool`

Default value: `false`

### **`cpuX.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`cpuX.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**cpuX.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**cpuX.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

### **`cpuX.min_sync_level`**

Force minimum `syncLevel` (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

### **`cpuX.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`cpuX.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`cpuX.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**cpuX.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**cpuX.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**cpuX.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**cpuX.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**cpuX.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**cpuX.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: 0x0

**cpuX.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: 0xf000000



**cpuX.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**cpuX.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**cpuX.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**cpuX.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**BROADCASTATOMIC**

Enable broadcasting of atomic operation. The `broadcastatomic` signal will override this value if used.

Type: `bool`

Default value: `true`

**BROADCASTACHEMAINT**

Enable broadcasting of cache maintenance operations to downstream caches. The `broadcastcachemaint` signal will override this value if used.

Type: `bool`

Default value: `true`

**BROADCASTOUTER**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

Type: `bool`

Default value: `true`

**BROADCASTPERSIST**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: `bool`

Default value: `true`

**CLUSTER\_ID**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the ManagerID64 of the core.

Type: `uint32_t`

Default value: `0x0`

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. This value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required.

Type: `uint8_t`

Default value: `1`

**GICDISABLE**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`

Default value: `true`

**NUM\_CORES**

Number of cores per cluster.

Type: `uint8_t`

Default value: 1

**cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: 0x1

**cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: 0x1

**dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**dcache-maintenance\_latency**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**dcache-miss\_latency**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**dcache-prefetch\_enabled**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

Type: `bool`

Default value: `false`

### **`dcache-read_access_latency`**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`dcache-read_latency`**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`dcache-snoop_data_transfer_latency`**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`dcache-state_modelled`**

Set whether D-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **`dcache-write_access_latency`**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **default\_opmode**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

Type: uint8\_t

Default value: 4

### **diagnostics**

Enable DynamIQ diagnostic messages.

Type: bool

Default value: false

### **enable\_simulation\_performance\_optimizations**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

Type: bool

Default value: true

### **ext\_abort\_device\_read\_is\_sync**

Synchronous reporting of device-nGnRE read external aborts.

Type: bool

Default value: false

### **ext\_abort\_device\_write\_is\_sync**

Synchronous reporting of device-nGnRE write external aborts.

Type: bool

Default value: `false`

**ext\_abort\_so\_read\_is\_sync**

Synchronous reporting of device-nGnRnE read external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_so\_write\_is\_sync**

Synchronous reporting of device-nGnRnE write external aborts.

Type: `bool`

Default value: `false`

**force\_zero\_PSTATE\_PAN**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

**has\_acp**

If true, Accelerator Coherency Port is configured.

Type: `bool`

Default value: `false`

**has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**has\_peripheral\_port**

If true, additional AXI peripheral port is configured.

Type: `bool`

Default value: `false`

### **has\_statistical\_profiling**

Whether Statistical Based Profiling is implemented (FEAT\_SPE).

Type: `bool`

Default value: `true`

### **icache-hit\_latency**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

### **icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and

intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`icache-read_latency`**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`icache-state_modelled`**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **`invalidate_code_cache_on_icache_cmo`**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: `uint8_t`

Default value: `0`

### **`l3cache-hit_latency`**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-maintenance_latency`**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`



**l3cache-miss\_latency**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-read\_access\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size `l3cache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-read\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l3cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-size**

L3 Cache size in bytes.

Type: `uint32_t`

Default value: `0x100000`

**l3cache-snoop\_data\_transfer\_latency**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-snoop\_issue\_latency**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-write_access_latency`**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size `l3cache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-write_latency`**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l3cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`pchannel_treat_simreset_as_poreset`**

Register core as ON state to cluster with simulation reset.

Type: `bool`

Default value: `false`

### **`periph_address_end`**

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

Type: `uint64_t`

Default value: `0x0`

### **`periph_address_start`**

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

Type: `uint64_t`

Default value: `0x0`

### **`ptw_latency`**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

### **tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

### **tlbi\_stall\_enabled**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: bool

Default value: false

### **treat-dcache-cmos-to-pou-as-nop**

Whether dcache invalidation to the point of unification is required for instruction to data coherence. 0 - Invalidate ops required, 1 - Invalidate ops not required and cannot generate faults, 2 - Invalidate ops not required but can generate faults.

Type: uint8\_t

Default value: 0

### **walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

## 3.76 ARMCortexA78AECT

Defined in LISA/ARMCortexA78AECT.lisa.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following parameters were added:

- `has_delayed_dbgreg`
- `has_delayed_sysreg`

## About ARMCortexA78AECT

The model supports the following features:

- DynamIQ (DSU) system registers.
- Per-core L2 cache.
- A P-Channel for the cluster and for each core.
- Optional peripheral port.
- L3 cache partition.
- Per-core clock.
- `BROADCASTPERSIST` pin.

The model does not support the following features:

- `BROADCASTCACHEMAINTPOU` pin.
- `COREINSTRRET`, `COREINSTRRUN`, or `NPMBIRQ` signals.
- DynamIQ features that are irrelevant to the programmers' view simulation, for example:
  - Automatic CPU retention mode.
  - Level-3 Cache RAM retention.
- 256-bit wide output transactions.
- ECC and parity schemes are hardware-specific so are not supported.
- Self-test features (MBIST).
- Latency configuration.
- Snoop filtering.
- Debug power management signals `DBGPWRDUP` and `DBGIRSTREQ` are not implemented, but `DBGPWRUPREQ` and `DBGNOPWRDWN` are implemented.
- Cache stashing capability.
- Split/Lock is supported but with the limitations described in the AE-specific features implemented section.

This model has a variable number of cores per cluster, specified using the `NUM_CORES` parameter.

The per-core parameters are preceded by `cpun.`, where `n` identifies the core (0-3).

This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (`crypto.dll` or `crypto.so`) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

## AE-specific features implemented

The only AE-specific features implemented in the model are Split-Lock and Lock-Step, with the following limitations:

- The implementation of Split-Lock and Lock-Step in the model is simply a halving of the number of cores available.
- Lock-step is implemented by a parameter that can be true or false for the entire execution only, rather than by a signal that can be enabled during execution.

As the model does not implement true Split-Lock functionality, the ports and parameters are contiguous. For example, for a 4-core CPU with Split-Lock enabled, `cpu0` and `cpu1` identify the available cores and associated ports, not `cpu0` and `cpu2`.

- Hybrid mode is not modeled in the DSU.

## Iris and MTI instances for ARMCortexA78AECT

This model has the following Iris instances:

Name	Instance type
ARMCortexA78AECT	Cluster_ARM_Cortex-A78AE
ARMCortexA78AECT.AMU	PVBusLogger
ARMCortexA78AECT.AMU.mapper	PVBusMapper
ARMCortexA78AECT.DAP	PVBusLogger
ARMCortexA78AECT.DAP.mapper	PVBusMapper
ARMCortexA78AECT.DSU	DSU
ARMCortexA78AECT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA78AECT.DSU.mpam_busslave	PVBusSlave
ARMCortexA78AECT.DSU.shared_cache	PVCache
ARMCortexA78AECT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA78AECT.DSU.shared_cache.upstream[Y] (where Y = 0-5)	PVBusSlave
ARMCortexA78AECT.MMAP	PVBusLogger
ARMCortexA78AECT.MMAP.mapper	PVBusMapper
ARMCortexA78AECT.RAS	PVBusLogger
ARMCortexA78AECT.RAS.mapper	PVBusMapper
ARMCortexA78AECT.cpuY (where Y = 0-1)	ARM_Cortex-A78AE
ARMCortexA78AECT.cpuY.UTLB (where Y = 0-1)	TLB
ARMCortexA78AECT.cpuY.debug_rom (where Y = 0-1)	debug_rom
ARMCortexA78AECT.cpuY.dtlb (where Y = 0-1)	TLB
ARMCortexA78AECT.cpuY.l1dcache (where Y = 0-1)	PVCache
ARMCortexA78AECT.cpuY.l1dcache.downstream[0].pvbusmaster (where Y = 0-1)	PVBusMaster
ARMCortexA78AECT.cpuY.l1dcache.upstream[0] (where Y = 0-1)	PVBusSlave
ARMCortexA78AECT.cpuY.l1licache (where Y = 0-1)	PVCache
ARMCortexA78AECT.cpuY.l1licache.downstream[0].pvbusmaster (where Y = 0-1)	PVBusMaster
ARMCortexA78AECT.cpuY.l1licache.upstream[0] (where Y = 0-1)	PVBusSlave

Name	Instance type
ARMCortexA78AECT.cpuY.l2cache (where Y = 0-1)	PVCache
ARMCortexA78AECT.cpuY.l2cache.downstream[0].pvbusmaster (where Y = 0-1)	PVBusMaster
ARMCortexA78AECT.cpuY.l2cache.upstream[U] (where Y = 0-1; U = 0-1)	PVBusSlave
ARMCortexA78AECT.ext_bus	PVBusLogger
ARMCortexA78AECT.ext_bus.mapper	PVBusMapper
ARMCortexA78AECT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA78AECT.global_debug_rom	debug_rom
ARMCortexA78AECT.secondary_debug_rom	debug_rom

This model has the following MTI trace components:

Name	Component type
ARMCortexA78AECT.AMU	PVBusLogger
ARMCortexA78AECT.AMU.mapper	PVBusMapper
ARMCortexA78AECT.DAP	PVBusLogger
ARMCortexA78AECT.DAP.mapper	PVBusMapper
ARMCortexA78AECT.DSU	DSU
ARMCortexA78AECT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA78AECT.DSU.mpam_busslave	PVBusSlave
ARMCortexA78AECT.DSU.shared_cache	PVCache
ARMCortexA78AECT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA78AECT.DSU.shared_cache.upstream[Y] (where Y = 0-5)	PVBusSlave
ARMCortexA78AECT.MMAP	PVBusLogger
ARMCortexA78AECT.MMAP.mapper	PVBusMapper
ARMCortexA78AECT.RAS	PVBusLogger
ARMCortexA78AECT.RAS.mapper	PVBusMapper
ARMCortexA78AECT.cpuY (where Y = 0-1)	ARM_Cortex-A78AE
ARMCortexA78AECT.cpuY.UTLB (where Y = 0-1)	TLB
ARMCortexA78AECT.cpuY.l1dcache (where Y = 0-1)	PVCache
ARMCortexA78AECT.cpuY.l1dcache.downstream[0].pvbusmaster (where Y = 0-1)	PVBusMaster
ARMCortexA78AECT.cpuY.l1dcache.upstream[0] (where Y = 0-1)	PVBusSlave
ARMCortexA78AECT.cpuY.l1icache (where Y = 0-1)	PVCache
ARMCortexA78AECT.cpuY.l1icache.downstream[0].pvbusmaster (where Y = 0-1)	PVBusMaster
ARMCortexA78AECT.cpuY.l1icache.upstream[0] (where Y = 0-1)	PVBusSlave
ARMCortexA78AECT.cpuY.l2cache (where Y = 0-1)	PVCache
ARMCortexA78AECT.cpuY.l2cache.downstream[0].pvbusmaster (where Y = 0-1)	PVBusMaster
ARMCortexA78AECT.cpuY.l2cache.upstream[U] (where Y = 0-1; U = 0-1)	PVBusSlave
ARMCortexA78AECT.ext_bus	PVBusLogger
ARMCortexA78AECT.ext_bus.mapper	PVBusMapper
ARMCortexA78AECT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

## Ports for ARMCortexA78AECT

Port	Direction	Protocol	Description
acp_s	slave	PVBus	AXI ACP slave port.
AENDMP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
ASTARTMP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	slave	Signal	CHI defined pins.
broadcastcachemaint	slave	Signal	ACE defined pins.
broadcastouter	slave	Signal	ACE defined pins.
broadcastpersist	slave	Signal	CHI defined pins.
cfgend	slave	Signal	This signal is for EE bit initialisation.
cfgsdisable	slave	Signal	This signal disables write access to some secure Interrupt Controller registers.
cfgte	slave	Signal	This signal provides default exception handling state.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
clusterid	slave	Value	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set ManagerID64 of the core.
clusterpmuirq	master	Signal	DynamiQ pmu irq
CNTHPIRQ	master	Signal	Timer signals to SOC.
CNTHVIRQ	master	Signal	Timer signals to SOC.
CNTPNSIRQ	master	Signal	Timer signals to SOC.
CNTPSIRQ	master	Signal	Timer signals to SOC.
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.
CNTVIRQ	master	Signal	Timer signals to SOC.
commirq	master	Signal	Interrupt signal from debug communications channel.
core_clk_in	slave	ClockSignal	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
cpuporeset	slave	Signal	CPU power on reset. Initializes all the processor logic, including debug logic.
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
ctidbgirq	master	Signal	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	slave	Signal	External debug interface.

Port	Direction	Protocol	Description
dbgnopwrdown	master	Signal	No power-down request.
dbgpwrupreq	master	Signal	Debug power up request.
dev_debug_s	slave	PVBus	External debug interface.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s	slave	GLCv3Comms	GLCv3 AXI-stream port.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	slave	PVBus	External debug interface.
niden	slave	Signal	External debug interface.
pchannel_cluster	slave	PChannel	PChannel for cluster.
pchannel_core	slave	PChannel	PChannels for cores
pmbirq	master	Signal	Interrupt signal from statistical profiling unit.
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	master	PVBus	The core will generate bus requests on this port.
pvbus_periph_m	master	PVBus	The core can generate peripheral bus request on this port.
rei	slave	Signal	Per core RAM Error Interrupt.
reset	slave	Signal	Raising this signal will put the core into reset mode.
rvbaraddr	slave	Value_64	Reset vector base address.
sci_m	master	SystemCoherencyInterface	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei	slave	Signal	Per core System Error physical pins.
spiden	slave	Signal	External debug interface.
spniden	slave	Signal	External debug interface.
sporeset	slave	Signal	A single cluster-wide power on reset signal for all resettable registers in DynamIQ.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfiq	slave	Signal	Virtualised FIQ.
vinithi	slave	Signal	This signal controls of the location of the exception vectors at reset.
virq	slave	Signal	Virtualised IRQ.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Per core virtual System Error physical pins.



## Parameters for ARMCortexA78AECT

### **cpuX.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **cpuX.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **cpuX.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **cpuX.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **cpuX.VINITHI**

Reset value of SCTLR.V.

Type: `bool`

Default value: `false`

### **cpuX.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**cpuX.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**cpuX.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**cpuX.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**cpuX.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**cpuX.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: 0xf000

**cpuX.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: 0x123456

**cpuX.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**cpuX.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**cpuX.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**cpuX.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**cpuX.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: true

**cpuX.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**cpuX.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**cpuX.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**cpuX.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**cpuX.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**cpuX.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**BROADCASTATOMIC**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`

Default value: `true`

**BROADCASTCACHEMAINT**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

Type: `bool`

Default value: `true`

**BROADCASTOUTER**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

Type: `bool`

Default value: `true`

**BROADCASTPERSIST**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: `bool`

Default value: `true`

**CLUSTER\_ID**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the ManagerID64 of the core.

Type: `uint32_t`

Default value: `0x0`

### **CMO\_broadcast\_when\_cache\_state\_modelling\_disabled**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. This value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required.

Type: `uint8_t`

Default value: `1`

### **GICDISABLE**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`

Default value: `true`

### **NUM\_CORES**

Number of cores per cluster.

Type: `uint8_t`

Default value: `2`

### **cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

### **cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

### **dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`dcache-maintenance_latency`**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`dcache-miss_latency`**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`dcache-prefetch_enabled`**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

Type: `bool`

Default value: `false`

### **`dcache-read_access_latency`**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`dcache-read_latency`**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`



**dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-state\_modelled**

Set whether D-cache has stateful implementation.

Type: `bool`

Default value: `false`

**dcache-write\_access\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**default\_opmode**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

Type: `uint8_t`

Default value: 4

**diagnostics**

Enable DynamIQ diagnostic messages.

Type: `bool`

Default value: `false`

**enable\_lock\_step**

Whether the core is configured in Dual Core Lock Step mode (FEAT\_DCLS).

Type: `bool`

Default value: `false`

**enable\_simulation\_performance\_optimizations**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of `stage12_tlb_size` parameter to 1024).

Type: `bool`

Default value: `true`

**ext\_abort\_device\_read\_is\_sync**

Synchronous reporting of device-nGnRE read external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_device\_write\_is\_sync**

Synchronous reporting of device-nGnRE write external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_so\_read\_is\_sync**

Synchronous reporting of device-nGnRnE read external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_so\_write\_is\_sync**

Synchronous reporting of device-nGnRnE write external aborts.

Type: `bool`

Default value: `false`

**force\_zero\_PSTATE\_PAN**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

**has\_acp**

If true, Accelerator Coherency Port is configured.

Type: `bool`

Default value: `false`

**has\_coherent\_icache**

Whether icache invalidation to the point of unification is required for instruction to data coherence. true - Invalidate operations not required.

Type: `bool`

Default value: `false`

**has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**has\_peripheral\_port**

If true, additional AXI peripheral port is configured.

Type: `bool`

Default value: `false`

**has\_statistical\_profiling**

Whether Statistical Based Profiling is implemented (FEAT\_SPE).

Type: `bool`

Default value: `true`

### **`icache-hit_latency`**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`icache-maintenance_latency`**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`icache-miss_latency`**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`icache-prefetch_enabled`**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

### **`icache-read_access_latency`**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-state\_modelled**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

**invalidate\_code\_cache\_on\_icache\_cmo**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: `uint8_t`

Default value: `0`

**l3cache-hit\_latency**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-maintenance\_latency**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-miss\_latency**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-read\_access\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size l3cache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**l3cache-read\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. l3cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**l3cache-size**

L3 Cache size in bytes.

Type: uint32\_t

Default value: 0x100000

**l3cache-snoop\_data\_transfer\_latency**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**l3cache-snoop\_issue\_latency**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**l3cache-write\_access\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size l3cache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used

instead of per-byte even if l3cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **l3cache-write\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. l3cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **pchannel\_treat\_simreset\_as\_poreset**

Register core as ON state to cluster with simulation reset.

Type: bool

Default value: false

### **periph\_address\_end**

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

Type: uint64\_t

Default value: 0x0

### **periph\_address\_start**

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

Type: uint64\_t

Default value: 0x0

### **ptw\_latency**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

### **reported\_patch\_level**

Purely cosmetic patch level value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: `int8_t`

Default value: `-1`

### **reported\_revision\_number**

Purely cosmetic revision number value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (`-1`) will see this parameter ignored.

Type: `int8_t`

Default value: `-1`

### **tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

### **tlbi\_stall\_enabled**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: `bool`

Default value: `false`

### **treat-dcache-cmos-to-pou-as-nop**

Whether dcache invalidation to the point of unification is required for instruction to data coherence. 0 - Invalidate ops required, 1 - Invalidate ops not required and cannot generate faults, 2 - Invalidate ops not required but can generate faults.

Type: `uint8_t`

Default value: `0`

### **treat\_PAC\_as\_NOP**

Non-architecture parameter. Treat Pointer Authentication as **NOP**. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are **NOP**. Branch, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

Type: `bool`

Default value: `false`

### **walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.



Type: `uint64_t`

Default value: `0x0`

## 3.77 ARMCortexA78CCT

Defined in `LISA/ARMCortexA78CCT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following parameters were added:

- `has_delayed_dbgreg`
- `has_delayed_sysreg`

### About ARMCortexA78CCT

The model supports the following features:

- DynamIQ (DSU) system registers.
- Per-core L2 cache.
- A P-Channel for the cluster and for each core.
- Optional peripheral port.
- L3 cache partition.
- Per-core clock.
- `BROADCASTPERSIST` pin.

The model does not support the following features:

- `BROADCASTCACHEMAINTPOU` pin.
- `COREINSTRRET`, `COREINSTRRUN`, or `nPMBIRQ` signals.
- DynamIQ features that are irrelevant to the programmers' view simulation, for example:
  - Automatic CPU retention mode.
  - Level-3 Cache RAM retention.
- 256-bit wide output transactions.
- ECC and parity schemes are hardware-specific so are not supported.
- Self-test features (MBIST).

- Latency configuration.
- Snoop filtering.
- Debug power management signals `DBGPWRDUP` and `DBGIRSTREQ` are not implemented, but `DBGPWRUPREQ` and `DBGNOPWRDWN` are implemented.
- Cache stashing capability.

This model has a variable number of cores per cluster, specified using the `NUM_CORES` parameter.

The per-core parameters are preceded by `cpun.`, where `n` identifies the core (0-7).

This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (`crypto.dll` or `crypto.so`) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

### Iris and MTI instances for ARMCortexA78CCT

This model has the following Iris instances:

Name	Instance type
ARMCortexA78CCT	Cluster_ARM_Cortex-A78C
ARMCortexA78CCT.AMU	PVBusLogger
ARMCortexA78CCT.AMU.mapper	PVBusMapper
ARMCortexA78CCT.DAP	PVBusLogger
ARMCortexA78CCT.DAP.mapper	PVBusMapper
ARMCortexA78CCT.DSU	DSU
ARMCortexA78CCT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA78CCT.DSU.mpam_busslave	PVBusSlave
ARMCortexA78CCT.DSU.shared_cache	PVCache
ARMCortexA78CCT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA78CCT.DSU.shared_cache.upstream[Y] (where Y = 0-3)	PVBusSlave
ARMCortexA78CCT.MMAP	PVBusLogger
ARMCortexA78CCT.MMAP.mapper	PVBusMapper
ARMCortexA78CCT.RAS	PVBusLogger
ARMCortexA78CCT.RAS.mapper	PVBusMapper
ARMCortexA78CCT.cpu0	ARM_Cortex-A78C
ARMCortexA78CCT.cpu0.UTLB	TLB
ARMCortexA78CCT.cpu0.debug_rom	debug_rom
ARMCortexA78CCT.cpu0.dtlb	TLB
ARMCortexA78CCT.cpu0.l1dcache	PVCache
ARMCortexA78CCT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA78CCT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA78CCT.cpu0.l1icache	PVCache
ARMCortexA78CCT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA78CCT.cpu0.l1icache.upstream[0]	PVBusSlave

Name	Instance type
ARMCortexA78CCT.cpu0.l2cache	PVCache
ARMCortexA78CCT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA78CCT.cpu0.l2cache.upstream[U] (where U = 0-1)	PVBusSlave
ARMCortexA78CCT.ext_bus	PVBusLogger
ARMCortexA78CCT.ext_bus.mapper	PVBusMapper
ARMCortexA78CCT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA78CCT.global_debug_rom	debug_rom
ARMCortexA78CCT.secondary_debug_rom	debug_rom

This model has the following MTI trace components:

Name	Component type
ARMCortexA78CCT.AMU	PVBusLogger
ARMCortexA78CCT.AMU.mapper	PVBusMapper
ARMCortexA78CCT.DAP	PVBusLogger
ARMCortexA78CCT.DAP.mapper	PVBusMapper
ARMCortexA78CCT.DSU	DSU
ARMCortexA78CCT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA78CCT.DSU.mpam_busslave	PVBusSlave
ARMCortexA78CCT.DSU.shared_cache	PVCache
ARMCortexA78CCT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA78CCT.DSU.shared_cache.upstream[Y] (where Y = 0-3)	PVBusSlave
ARMCortexA78CCT.MMAP	PVBusLogger
ARMCortexA78CCT.MMAP.mapper	PVBusMapper
ARMCortexA78CCT.RAS	PVBusLogger
ARMCortexA78CCT.RAS.mapper	PVBusMapper
ARMCortexA78CCT.cpu0	ARM_Cortex-A78C
ARMCortexA78CCT.cpu0.UTLB	TLB
ARMCortexA78CCT.cpu0.l1dcache	PVCache
ARMCortexA78CCT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA78CCT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA78CCT.cpu0.l1icache	PVCache
ARMCortexA78CCT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA78CCT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA78CCT.cpu0.l2cache	PVCache
ARMCortexA78CCT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA78CCT.cpu0.l2cache.upstream[U] (where U = 0-1)	PVBusSlave
ARMCortexA78CCT.ext_bus	PVBusLogger
ARMCortexA78CCT.ext_bus.mapper	PVBusMapper
ARMCortexA78CCT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

## Ports for ARMCortexA78CCT

Port	Direction	Protocol	Description
acp_s	slave	PVBus	AXI ACP slave port.
AENDMP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
ASTARTMP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	slave	Signal	CHI defined pins.
broadcastcachemaint	slave	Signal	ACE defined pins.
broadcastouter	slave	Signal	ACE defined pins.
broadcastpersist	slave	Signal	CHI defined pins.
cfgend	slave	Signal	This signal is for EE bit initialisation.
cfgsdisable	slave	Signal	This signal disables write access to some secure Interrupt Controller registers.
cfgte	slave	Signal	This signal provides default exception handling state.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
clusterid	slave	Value	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set ManagerID64 of the core.
clusterpmuirq	master	Signal	DynamiQ pmu irq
CNTHPIRQ	master	Signal	Timer signals to SOC.
CNTHVIRQ	master	Signal	Timer signals to SOC.
CNTPNSIRQ	master	Signal	Timer signals to SOC.
CNTPSIRQ	master	Signal	Timer signals to SOC.
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.
CNTVIRQ	master	Signal	Timer signals to SOC.
commirq	master	Signal	Interrupt signal from debug communications channel.
core_clk_in	slave	ClockSignal	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
cpuporeset	slave	Signal	CPU power on reset. Initializes all the processor logic, including debug logic.
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
ctidbgirq	master	Signal	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	slave	Signal	External debug interface.

Port	Direction	Protocol	Description
dbgnopwrdown	master	Signal	No power-down request.
dbgpwrupreq	master	Signal	Debug power up request.
dev_debug_s	slave	PVBus	External debug interface.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	slave	PVBus	External debug interface.
niden	slave	Signal	External debug interface.
pchannel_cluster	slave	PChannel	PChannel for cluster.
pchannel_core	slave	PChannel	PChannels for cores
pmbirq	master	Signal	Interrupt signal from the statistical profiling unit.
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	master	PVBus	The core will generate bus requests on this port.
pvbus_periph_m	master	PVBus	The core can generate peripheral bus request on this port.
rei	slave	Signal	Per core RAM Error Interrupt.
reset	slave	Signal	Raising this signal will put the core into reset mode.
rvbaraddr	slave	Value_64	Reset vector base address.
sci_m	master	SystemCoherencyInterface	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei	slave	Signal	Per core System Error physical pins.
spiden	slave	Signal	External debug interface.
spniden	slave	Signal	External debug interface.
sporeset	slave	Signal	A single cluster-wide power on reset signal for all resettable registers in DynamiQ.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfiq	slave	Signal	Virtualised FIQ.
vinithi	slave	Signal	This signal controls of the location of the exception vectors at reset.
virq	slave	Signal	Virtualised IRQ.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Per core virtual System Error physical pins.

## Parameters for ARMCortexA78CCT

### **cpuX.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **cpuX.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **cpuX.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **cpuX.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **cpuX.VINITHI**

Reset value of SCTLR.V.

Type: `bool`

Default value: `false`

### **cpuX.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**cpuX.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**cpuX.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**cpuX.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**cpuX.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**cpuX.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**cpuX.l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: 0x80000

**cpuX.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**cpuX.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t



Default value: 0

**cpuX.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**cpuX.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**cpuX.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**cpuX.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**cpuX.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**cpuX.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**cpuX.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**cpuX.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: true

**cpuX.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**cpuX.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**cpuX.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**cpuX.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**cpuX.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**cpuX.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**BROADCASTATOMIC**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`

Default value: `true`

**BROADCASTCACHEMAINT**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

Type: `bool`

Default value: `true`

**BROADCASTOUTER**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

Type: `bool`

Default value: `true`

**BROADCASTPERSIST**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: `bool`

Default value: `true`

**CLUSTER\_ID**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the ManagerID64 of the core.

Type: `uint32_t`

Default value: `0x0`

### **CMO\_broadcast\_when\_cache\_state\_modelling\_disabled**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. This value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required.

Type: `uint8_t`

Default value: `1`

### **GICDISABLE**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`

Default value: `true`

### **NUM\_CORES**

Number of cores per cluster.

Type: `uint8_t`

Default value: `1`

### **cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

### **cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

### **dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`dcache-maintenance_latency`**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`dcache-miss_latency`**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`dcache-prefetch_enabled`**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

Type: `bool`

Default value: `false`

### **`dcache-read_access_latency`**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`dcache-read_latency`**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-size**

L1 D-Cache size in bytes.

Type: `uint32_t`

Default value: `0x10000`

**dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-state\_modelled**

Set whether D-cache has stateful implementation.

Type: `bool`

Default value: `false`

**dcache-write\_access\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**default\_opmode**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

Type: `uint8_t`

Default value: 4

**diagnostics**

Enable DynaMIQ diagnostic messages.

Type: `bool`

Default value: `false`

**enable\_simulation\_performance\_optimizations**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of `stage12_tlb_size` parameter to 1024).

Type: `bool`

Default value: `true`

**ext\_abort\_device\_read\_is\_sync**

Synchronous reporting of device-nGnRE read external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_device\_write\_is\_sync**

Synchronous reporting of device-nGnRE write external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_so\_read\_is\_sync**

Synchronous reporting of device-nGnRnE read external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_so\_write\_is\_sync**

Synchronous reporting of device-nGnRnE write external aborts.

Type: `bool`

Default value: `false`

**force\_zero\_PSTATE\_PAN**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

**has\_acp**

If true, Accelerator Coherency Port is configured.

Type: `bool`

Default value: `false`

**has\_coherent\_icache**

Whether icache invalidation to the point of unification is required for instruction to data coherence. true - Invalidate operations not required.

Type: `bool`

Default value: `false`

**has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**has\_dot\_product**

Possible values of this parameter are:- 1, feature not implemented.- 2, feature is implemented.

Type: `uint8_t`

Default value: 2

**has\_peripheral\_port**

If true, additional AXI peripheral port is configured.



Type: `bool`

Default value: `false`

### **`icache-hit_latency`**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`icache-maintenance_latency`**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`icache-miss_latency`**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`icache-prefetch_enabled`**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

### **`icache-read_access_latency`**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-size**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: `0x10000`

**icache-state\_modelled**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

**invalidate\_code\_cache\_on\_icache\_cmo**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: `uint8_t`

Default value: `0`

**l3cache-hit\_latency**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-maintenance\_latency**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-miss\_latency**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-read\_access\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size `l3cache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-read\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l3cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-size**

L3 Cache size in bytes.

Type: `uint32_t`

Default value: `0x100000`

**l3cache-snoop\_data\_transfer\_latency**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-snoop\_issue\_latency**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-write_access_latency`**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size `l3cache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-write_latency`**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l3cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`pchannel_treat_simreset_as_poreset`**

Register core as ON state to cluster with simulation reset.

Type: `bool`

Default value: `false`

### **`periph_address_end`**

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

Type: `uint64_t`

Default value: `0x0`

### **`periph_address_start`**

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

Type: `uint64_t`

Default value: `0x0`

### **`ptw_latency`**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

### **tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

### **tlbi\_stall\_enabled**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: bool

Default value: false

### **treat-dcache-cmos-to-pou-as-nop**

Whether dcache invalidation to the point of unification is required for instruction to data coherence. 0 - Invalidate ops required, 1 - Invalidate ops not required and cannot generate faults, 2 - Invalidate ops not required but can generate faults.

Type: uint8\_t

Default value: 0

### **treat\_PAC\_as\_NOP**

Non-architecture parameter. Treat Pointer Authentication as **NOP**. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are **NOP**. Branch, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

Type: bool

Default value: false

### **walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

## 3.78 ARMCortexA78CT

Defined in `LISA/ARMCortexA78CT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following parameters were added:

- `has_delayed_dbgreg`
- `has_delayed_sysreg`

### About ARMCortexA78CT

The model supports the following features:

- DynamIQ (DSU) system registers.
- Per-core L2 cache.
- A P-Channel for the cluster and for each core.
- Optional peripheral port.
- L3 cache partition.
- Per-core clock.
- `BROADCASTPERSIST` pin.

The model does not support the following features:

- `BROADCASTCACHEMAINTPOU` pin.
- `COREINSTRRET`, `COREINSTRRUN`, or `nPMBIRQ` signals.
- DynamIQ features that are irrelevant to the programmers' view simulation, for example:
  - Automatic CPU retention mode.
  - Level-3 Cache RAM retention.
- 256-bit wide output transactions.
- ECC and parity schemes are hardware-specific so are not supported.
- Self-test features (MBIST).
- Latency configuration.
- Snoop filtering.
- Debug power management signals `DBGPWRDUP` and `DBGIRSTREQ` are not implemented, but `DBGPWRUPREQ` and `DBGNOPWRDWN` are implemented.

- Cache stashing capability.

This model has a variable number of cores per cluster, specified using the `NUM_CORES` parameter.

The per-core parameters are preceded by `cpun.`, where `n` identifies the core (0-3).

This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (`crypto.dll` or `crypto.so`) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

## Iris and MTI instances for ARM Cortex A78CT

This model has the following Iris instances:

Name	Instance type
ARMCortexA78CT	Cluster_ARM_Cortex-A78
ARMCortexA78CT.AMU	PVBusLogger
ARMCortexA78CT.AMU.mapper	PVBusMapper
ARMCortexA78CT.DAP	PVBusLogger
ARMCortexA78CT.DAP.mapper	PVBusMapper
ARMCortexA78CT.DSU	DSU
ARMCortexA78CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA78CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA78CT.DSU.shared_cache	PVCache
ARMCortexA78CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA78CT.DSU.shared_cache.upstream[Y] (where Y = 0-3)	PVBusSlave
ARMCortexA78CT.MMAP	PVBusLogger
ARMCortexA78CT.MMAP.mapper	PVBusMapper
ARMCortexA78CT.RAS	PVBusLogger
ARMCortexA78CT.RAS.mapper	PVBusMapper
ARMCortexA78CT.cpu0	ARM_Cortex-A78
ARMCortexA78CT.cpu0.UTLB	TLB
ARMCortexA78CT.cpu0.debug_rom	debug_rom
ARMCortexA78CT.cpu0.dtlb	TLB
ARMCortexA78CT.cpu0.l1dcache	PVCache
ARMCortexA78CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA78CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA78CT.cpu0.l1icache	PVCache
ARMCortexA78CT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA78CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA78CT.cpu0.l2cache	PVCache
ARMCortexA78CT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA78CT.cpu0.l2cache.upstream[U] (where U = 0-1)	PVBusSlave
ARMCortexA78CT.ext_bus	PVBusLogger

Name	Instance type
ARMCortexA78CT.ext_bus.mapper	PVBusMapper
ARMCortexA78CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexA78CT.global_debug_rom	debug_rom
ARMCortexA78CT.secondary_debug_rom	debug_rom

This model has the following MTI trace components:

Name	Component type
ARMCortexA78CT.AMU	PVBusLogger
ARMCortexA78CT.AMU.mapper	PVBusMapper
ARMCortexA78CT.DAP	PVBusLogger
ARMCortexA78CT.DAP.mapper	PVBusMapper
ARMCortexA78CT.DSU	DSU
ARMCortexA78CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexA78CT.DSU.mpam_busslave	PVBusSlave
ARMCortexA78CT.DSU.shared_cache	PVCache
ARMCortexA78CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA78CT.DSU.shared_cache.upstream[Y] (where Y = 0–3)	PVBusSlave
ARMCortexA78CT.MMAP	PVBusLogger
ARMCortexA78CT.MMAP.mapper	PVBusMapper
ARMCortexA78CT.RAS	PVBusLogger
ARMCortexA78CT.RAS.mapper	PVBusMapper
ARMCortexA78CT.cpu0	ARM_Cortex-A78
ARMCortexA78CT.cpu0.UTLB	TLB
ARMCortexA78CT.cpu0.l1dcache	PVCache
ARMCortexA78CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA78CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexA78CT.cpu0.l1icache	PVCache
ARMCortexA78CT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA78CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexA78CT.cpu0.l2cache	PVCache
ARMCortexA78CT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexA78CT.cpu0.l2cache.upstream[U] (where U = 0–1)	PVBusSlave
ARMCortexA78CT.ext_bus	PVBusLogger
ARMCortexA78CT.ext_bus.mapper	PVBusMapper
ARMCortexA78CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

## Ports for ARMCortexA78CT

Port	Direction	Protocol	Description
acp_s	slave	PVBus	AXI ACP slave port.



Port	Direction	Protocol	Description
AENDMP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
ASTARTMP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	slave	Signal	CHI defined pins.
broadcastcachemaint	slave	Signal	ACE defined pins.
broadcastouter	slave	Signal	ACE defined pins.
broadcastpersist	slave	Signal	CHI defined pins.
cfgend	slave	Signal	This signal if for EE bit initialisation.
cfgsdisable	slave	Signal	This signal disables write access to some secure Interrupt Controller registers.
cfgte	slave	Signal	This signal provides default exception handling state.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
clusterid	slave	Value	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set ManagerID64 of the core.
clusterpmuirq	master	Signal	DynamiQ pmu irq
CNTHPIRQ	master	Signal	Timer signals to SOC.
CNTHVIRQ	master	Signal	Timer signals to SOC.
CNTPNSIRQ	master	Signal	Timer signals to SOC.
CNTPSIRQ	master	Signal	Timer signals to SOC.
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.
CNTVIRQ	master	Signal	Timer signals to SOC.
commirq	master	Signal	Interrupt signal from debug communications channel.
core_clk_in	slave	ClockSignal	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
cpuporeset	slave	Signal	CPU power on reset. Initializes all the processor logic, including debug logic.
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
ctidbgirq	master	Signal	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	slave	Signal	External debug interface.
dbgnopwrdown	master	Signal	No power-down request.
dbgprupreq	master	Signal	Debug power up request.
dev_debug_s	slave	PVBus	External debug interface.

Port	Direction	Protocol	Description
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	slave	PVBus	External debug interface.
niden	slave	Signal	External debug interface.
pchannel_cluster	slave	PChannel	PChannel for cluster.
pchannel_core	slave	PChannel	PChannels for cores
pmbirq	master	Signal	Interrupt signal from the statistical profiling unit.
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	master	PVBus	The core will generate bus requests on this port.
pvbus_periph_m	master	PVBus	The core can generate peripheral bus request on this port.
rei	slave	Signal	Per core RAM Error Interrupt.
reset	slave	Signal	Raising this signal will put the core into reset mode.
rvbaraddr	slave	Value_64	Reset vector base address.
sci_m	master	SystemCoherencyInterface	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei	slave	Signal	Per core System Error physical pins.
spiden	slave	Signal	External debug interface.
spniden	slave	Signal	External debug interface.
sporeset	slave	Signal	A single cluster-wide power on reset signal for all resettable registers in DynamIQ.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfiq	slave	Signal	Virtualised FIQ.
vinithi	slave	Signal	This signal controls of the location of the exception vectors at reset.
virq	slave	Signal	Virtualised IRQ.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Per core virtual System Error physical pins.

## Parameters for ARMCortexA78CT

### **cpuX.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **cpuX.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **cpuX.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **cpuX.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **cpuX.VINITHI**

Reset value of SCTLR.V.

Type: `bool`

Default value: `false`

### **cpuX.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**cpuX.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**cpuX.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**cpuX.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**cpuX.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**cpuX.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**cpuX.l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: 0x80000

**cpuX.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**cpuX.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**cpuX.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**cpuX.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**cpuX.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**cpuX.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**cpuX.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**cpuX.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**cpuX.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**cpuX.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: true

**cpuX.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**cpuX.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**cpuX.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**cpuX.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**cpuX.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**cpuX.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**BROADCASTATOMIC**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`

Default value: `true`

**BROADCASTCACHEMAINT**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

Type: `bool`

Default value: `true`

**BROADCASTOUTER**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

Type: `bool`

Default value: `true`

**BROADCASTPERSIST**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: `bool`

Default value: `true`

**CLUSTER\_ID**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the ManagerID64 of the core.



Type: `uint32_t`

Default value: `0x0`

### **CMO\_broadcast\_when\_cache\_state\_modelling\_disabled**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. This value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required.

Type: `uint8_t`

Default value: `1`

### **GICDISABLE**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`

Default value: `true`

### **NUM\_CORES**

Number of cores per cluster.

Type: `uint8_t`

Default value: `1`

### **cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

### **cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

### **dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`dcache-maintenance_latency`**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`dcache-miss_latency`**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`dcache-prefetch_enabled`**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

Type: `bool`

Default value: `false`

### **`dcache-read_access_latency`**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`dcache-read_latency`**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-size**

L1 D-Cache size in bytes.

Type: `uint32_t`

Default value: `0x10000`

**dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-state\_modelled**

Set whether D-cache has stateful implementation.

Type: `bool`

Default value: `false`

**dcache-write\_access\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**default\_opmode**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

Type: `uint8_t`

Default value: 4

**diagnostics**

Enable DynaMiq diagnostic messages.

Type: `bool`

Default value: `false`

**enable\_simulation\_performance\_optimizations**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of `stage12_tlb_size` parameter to 1024).

Type: `bool`

Default value: `true`

**ext\_abort\_device\_read\_is\_sync**

Synchronous reporting of device-nGnRE read external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_device\_write\_is\_sync**

Synchronous reporting of device-nGnRE write external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_so\_read\_is\_sync**

Synchronous reporting of device-nGnRnE read external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_so\_write\_is\_sync**

Synchronous reporting of device-nGnRnE write external aborts.

Type: `bool`

Default value: `false`

**force\_zero\_PSTATE\_PAN**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

**has\_acp**

If true, Accelerator Coherency Port is configured.

Type: `bool`

Default value: `false`

**has\_coherent\_icache**

Whether icache invalidation to the point of unification is required for instruction to data coherence. true - Invalidate operations not required.

Type: `bool`

Default value: `false`

**has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**has\_dot\_product**

Possible values of this parameter are:- 1, feature not implemented.- 2, feature is implemented.

Type: `uint8_t`

Default value: 2

**has\_peripheral\_port**

If true, additional AXI peripheral port is configured.

Type: `bool`

Default value: `false`

### **`icache-hit_latency`**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`icache-maintenance_latency`**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`icache-miss_latency`**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`icache-prefetch_enabled`**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

### **`icache-read_access_latency`**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-size**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: `0x10000`

**icache-state\_modelled**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

**invalidate\_code\_cache\_on\_icache\_cmo**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: `uint8_t`

Default value: `0`

**l3cache-hit\_latency**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-maintenance\_latency**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-miss\_latency**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-read\_access\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size `l3cache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-read\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l3cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-size**

L3 Cache size in bytes.

Type: `uint32_t`

Default value: `0x100000`

**l3cache-snoop\_data\_transfer\_latency**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-snoop\_issue\_latency**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.



Type: `uint64_t`

Default value: `0x0`

### **`l3cache-write_access_latency`**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size `l3cache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-write_latency`**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l3cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`pchannel_treat_simreset_as_poreset`**

Register core as ON state to cluster with simulation reset.

Type: `bool`

Default value: `false`

### **`periph_address_end`**

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

Type: `uint64_t`

Default value: `0x0`

### **`periph_address_start`**

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

Type: `uint64_t`

Default value: `0x0`

### **`ptw_latency`**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

### **tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

### **tlbi\_stall\_enabled**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: bool

Default value: false

### **treat-dcache-cmos-to-pou-as-nop**

Whether dcache invalidation to the point of unification is required for instruction to data coherence. 0 - Invalidate ops required, 1 - Invalidate ops not required and cannot generate faults, 2 - Invalidate ops not required but can generate faults.

Type: uint8\_t

Default value: 0

### **walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

## 3.79 ARMCortexM0CT

Defined in LISA/ARMCortexM0CT.lisa.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following ports were removed:

- `currpri`

## Differences between the model and the RTL

This model does not have a parameter that is equivalent to the RAR integration option. The architecturally-required register state is reset.

This model exposes a VTOR register through Iris but this register does not exist in the IP.

Armv6-M is a subset of Armv7-M. Arm does not guarantee that all Armv7-M-specific behavior is absent from Armv6-M Fast Models cores. Therefore, Arm does not guarantee that code that runs on Armv7-M cores but fails on Armv6-M cores will also fail on Armv6-M Fast Models cores.

## Iris and MTI instances for ARMCortexM0CT

This model has the following Iris instances:

Name	Instance type
ARMCortexM0CT	ARM_Cortex-M0
ARMCortexM0CT.acp_mapper	PVBusMapper
ARMCortexM0CT.ext_bus	PVBusLogger
ARMCortexM0CT.ext_bus.mapper	PVBusMapper
ARMCortexM0CT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

Name	Component type
ARMCortexM0CT	ARM_Cortex-M0
ARMCortexM0CT.acp_mapper	PVBusMapper
ARMCortexM0CT.ext_bus	PVBusLogger
ARMCortexM0CT.ext_bus.mapper	PVBusMapper
ARMCortexM0CT.l2_flusher	AsyncCacheFlushUnit

## Ports for ARMCortexM0CT

Port	Direction	Protocol	Description
ahbd	slave	PVBus	Debug AHB - core bus slave driven by the DAP.
bigend	slave	Signal	Configure big endian data format.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
edbgrq	slave	Signal	External debug request.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE and corresponds to the RTL TXEV and RXEV signals.
intisr	slave	Signal	This signal array delivers signals to the NVIC.
intnmi	slave	Signal	Configure non maskable interrupt.
lockup	master	Signal	Asserted when the processor is in lockup state.
poreset	slave	Signal	Raising this signal will do a power-on reset of the core.

Port	Direction	Protocol	Description
pv_ppbus_m	master	PVBus	The core will generate External Private Peripheral Bus requests on this port.
pvbus_m	master	PVBus	The core will generate bus requests on this port.
sleepdeep	master	Signal	Asserted when the processor is in deep sleep.
sleeping	master	Signal	Asserted when the processor is in sleep.
stcalib	slave	Value	This is the calibration value for the SysTick timer The following bit flag is modelled stcalib[25] - NOREF - Indicates no reference clock integrated. If stclk is not in use, set to 1
stclk	slave	ClockSignal	This is the reference clock for the SysTick timer, called STCLKEN in the specification
sysreset	slave	Signal	Raising this signal will put the core into reset mode (but does not reset the debug logic).
sysresetreq	master	Signal	Asserted to indicate that a reset is required.
ticks	master	InstructionCount	Port allowing the number of instructions since startup to be read from the CPU.

## Parameters for ARM Cortex M0CT

### **BIGENDINIT**

Initialize processor to big endian mode.

Type: `bool`

Default value: `false`

### **BKPT**

Number of breakpoint unit comparators implemented.

Type: `uint8_t`

Default value: 4

### **DBG**

Set whether debug extensions are implemented.

Type: `bool`

Default value: `true`

### **NUM\_IRQ**

Number of user interrupts.

Type: `uint8_t`

Default value: 32

### **SYST**

Enable support for SysTick timer functionality.

Type: `bool`

Default value: `true`

### **WIC**

Include support for WIC-mode deep sleep.

Type: `bool`

Default value: `true`

### **WPT**

Number of watchpoint unit comparators implemented.

Type: `uint8_t`

Default value: 2

### **cpi\_div**

divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

### **cpi\_mul**

multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

### **manager\_id**

Manager ID presented in bus transactions.

Type: `uint64_t`

Default value: `0x0`

### **min\_sync\_level**

force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: 0

### **reported\_patch\_level**

Purely cosmetic patch level value to be displayed from the CUID register. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: `int8_t`

Default value: -1

### **reported\_revision\_number**

Purely cosmetic revision number value to be displayed from the CPUID register. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: `int8_t`

Default value: -1

### **semihosting-Thumb\_SVC**

T32 SVC number for semihosting.

Type: `uint8_t`

Default value: 171

### **semihosting-cmd\_line**

Command line available to semihosting SVC calls.

Type: `string`

Default value: N/A

### **semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **semihosting-enable**

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

Type: `bool`

Default value: `true`

### **semihosting-heap\_base**

Virtual address of heap base.

Type: `uint32_t`

Default value: 0x0

**semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint32_t`

Default value: `0x20700000`

**semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint32_t`

Default value: `0x20800000`

**semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint32_t`

Default value: `0x20700000`

## 3.80 ARMCortexM0PlusCT

Defined in `LISA/ARMCortexM0PlusCT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p1	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following ports were removed:

- `currpri`

### Differences between the model and the RTL

This model does not have a parameter that is equivalent to the RAR integration option. The architecturally-required register state is reset.

This model exposes a VTOR register through Iris but this register does not exist in the IP.

Armv6-M is a subset of Armv7-M. Arm does not guarantee that all Armv7-M-specific behavior is absent from Armv6-M Fast Models cores. Therefore, Arm does not guarantee that code that runs on Armv7-M cores but fails on Armv6-M cores will also fail on Armv6-M Fast Models cores.

## Iris and MTI instances for ARMCortexM0PlusCT

This model has the following Iris instances:

Name	Instance type
ARMCortexM0PlusCT	ARM_Cortex-M0+
ARMCortexM0PlusCT.acp_mapper	PVBusMapper
ARMCortexM0PlusCT.ext_bus	PVBusLogger
ARMCortexM0PlusCT.ext_bus.mapper	PVBusMapper
ARMCortexM0PlusCT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

Name	Component type
ARMCortexM0PlusCT	ARM_Cortex-M0+
ARMCortexM0PlusCT.acp_mapper	PVBusMapper
ARMCortexM0PlusCT.ext_bus	PVBusLogger
ARMCortexM0PlusCT.ext_bus.mapper	PVBusMapper
ARMCortexM0PlusCT.l2_flusher	AsyncCacheFlushUnit

## Ports for ARMCortexM0PlusCT

Port	Direction	Protocol	Description
ahbd	slave	PVBus	Debug AHB - core bus slave driven by the DAP.
bigend	slave	Signal	Configure big endian data format.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
cpuwait	slave	Signal	CPUWAIT extends effect of reset when true
edbgrq	slave	Signal	External debug request.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE and corresponds to the RTL TXEV and RXEV signals.
intisr	slave	Signal	This signal array delivers signals to the NVIC.
intnmi	slave	Signal	Configure non maskable interrupt.
io_port_in	slave	PVBus	I/O port pair. See the documentation for the io_port_out port.
io_port_out	master	PVBus	I/O port pair. Used if IOP is true. Transactions from io_port_out which do not “match” should be returned via io_port_in. For performance reasons, the I/O port interface is not modelled directly. Instead, a simple PVBus gasket is inserted at the point in the memory system where the I/O port would be. In hardware, a device would be attached to the port which would tell the CPU whether it would like to intercept each transaction, given its address. This can be modelled in a performant manner by connecting a PVBusMapper-based device to io_port_out which intercepts transactions of interest and passes other transactions back to the CPU via io_port_in. Your I/O port device model is also responsible for aborting transactions which would be aborted on hardware (e.g. exclusives) if necessary.
lockup	master	Signal	Asserted when the processor is in lockup state.
poreset	slave	Signal	Raising this signal will do a power-on reset of the core.
pv_ppbus_m	master	PVBus	The core will generate External Private Peripheral Bus requests on this port.
pdbus_m	master	PVBus	The core will generate bus requests on this port.



Port	Direction	Protocol	Description
sleepdeep	master	Signal	Asserted when the processor is in deep sleep.
sleeping	master	Signal	Asserted when the processor is in sleep.
stcalib	slave	Value	This is the calibration value for the SysTick timer. The following bit flag is modelled: stcalib[25] - NOREF - Indicates no reference clock integrated. If stclk is not in use, set to 1.
stclk	slave	ClockSignal	This is the reference clock for the SysTick timer, called STCLKEN in the specification.
sysreset	slave	Signal	Raising this signal will put the core into reset mode (but does not reset the debug logic).
sysresetreq	master	Signal	Asserted to indicate that a reset is required.
ticks	master	InstructionCount	Port allowing the number of instructions since startup to be read from the CPU.

## Parameters for ARMCortexM0PlusCT

### **BIGENDINIT**

Initialize processor to big endian mode.

Type: `bool`

Default value: `false`

### **BKPT**

Number of breakpoint unit comparators implemented.

Type: `uint8_t`

Default value: `4`

### **DBG**

Set whether debug extensions are implemented.

Type: `bool`

Default value: `true`

### **IOP**

Send all d-side transactions to the port, `io_port_out`. Transactions which do not match should be returned to the port, `io_port_in`.

Type: `bool`

Default value: `false`

### **IRQDIS**

IRQ line disable mask. Bit `n` of this 32-bit parameter disables `IRQ[n]`.

Type: `uint32_t`

Default value: `0x0`

**NUM\_IRQ**

Number of user interrupts.

Type: `uint8_t`

Default value: 32

**NUM\_MPU\_REGION**

Number of MPU regions.

Type: `uint8_t`

Default value: 0

**SYST**

Enable support for SysTick timer functionality.

Type: `bool`

Default value: `true`

**USER**

Enable support for Unprivileged/Privileged Extension.

Type: `bool`

Default value: `false`

**VTOR**

Include Vector Table Offset Register.

Type: `bool`

Default value: `false`

**WIC**

Include support for WIC-mode deep sleep.

Type: `bool`

Default value: `true`

**WPT**

Number of watchpoint unit comparators implemented.

Type: `uint8_t`

Default value: 2

**cpi\_div**

divider for calculating CPI (Cycles Per Instruction).

Type: uint32\_t

Default value: 0x1

**cpi\_mul**

multiplier for calculating CPI (Cycles Per Instruction).

Type: uint32\_t

Default value: 0x1

**manager\_id**

Manager ID presented in bus transactions.

Type: uint64\_t

Default value: 0x0

**min\_sync\_level**

force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**reported\_patch\_level**

Purely cosmetic patch level value to be displayed from the CPUID register. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: int8\_t

Default value: -1

**reported\_revision\_number**

Purely cosmetic revision number value to be displayed from the CPUID register. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: int8\_t

Default value: -1

**semihosting-Thumb\_SVC**

T32 SVC number for semihosting.

Type: `uint8_t`

Default value: 171

### **semihosting-cmd\_line**

Command line available to semihosting SVC calls.

Type: `string`

Default value: N/A

### **semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **semihosting-enable**

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

Type: `bool`

Default value: `true`

### **semihosting-heap\_base**

Virtual address of heap base.

Type: `uint32_t`

Default value: `0x0`

### **semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint32_t`

Default value: `0x20700000`

### **semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint32_t`

Default value: `0x20800000`

**semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint32_t`

Default value: `0x20700000`

## 3.81 ARMCortexM23CT

Defined in `LISA/ARMCortexM23CT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r1p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following ports were removed:

- `currpri`

### Differences between the CT model and RTL implementations

The model does not support MTB, ETM, or TPIU. MTB RAM is absent on the model.

### Iris and MTI instances for ARMCortexM23CT

This model has the following Iris instances:

Name	Instance type
ARMCortexM23CT	ARM_Cortex-M23
ARMCortexM23CT.acp_mapper	PVBusMapper
ARMCortexM23CT.ext_bus	PVBusLogger
ARMCortexM23CT.ext_bus.mapper	PVBusMapper
ARMCortexM23CT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

Name	Component type
ARMCortexM23CT	ARM_Cortex-M23
ARMCortexM23CT.acp_mapper	PVBusMapper
ARMCortexM23CT.ext_bus	PVBusLogger
ARMCortexM23CT.ext_bus.mapper	PVBusMapper
ARMCortexM23CT.l2_flusher	AsyncCacheFlushUnit

## Ports for ARMCortexM23CT

Port	Direction	Protocol	Description
ahbd	slave	PVBus	Debug AHB - core bus slave driven by the DAP.
bigend	slave	Signal	Configure big endian data format.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
core_dside_bus_gasket_in	slave	PVBus	Inbound D-side PVBus link is used when the external gasket is enabled. This allows externally processed D-side traffic to be fed back into the core path.
core_dside_bus_gasket_out	master	PVBus	Outbound D-side PVBus is exposed when has_core_dside_bus_gasket is true. This lets an external gasket observe or alter core data traffic before it re-enters on the inbound port.
cpuwait	slave	Signal	Clear = Core goes through reset sequence as normal, Set = Core waits out of reset.
dap_s	slave	PVBus	Debug Access Port (DAP).
dbgen	slave	Signal	Invasive debug control signals. Debug enable, Set=enabled, Clear=disabled
dbgrestart	slave	Signal	External request to leave debug state
dbgrestarted	master	Signal	Acknowledge for DBGRESTART
edbgrq	slave	Signal	External debug request.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE and corresponds to the RTL TXEV and RXEV signals.
halted	master	Signal	Core is in halt mode debug state
hreset	slave	Signal	Raising this signal will put the core into reset mode (but does not reset the debug logic).
idau_invalidate_region	slave	Value_64	64 bit number to invalid IDAU memory range (start_address<<32 end_address)
idau	master	PVBus	The core will generate IDAU requests on this port.
initvtor	slave	Value	Reset configuration port - Secure Vector table offset (VTOR.TBLOFF[31:7]) out of reset This port remains functional no matter ARMv8-M Security Extensions are included or not When ARMv8-M Security Extensions are not included, all exceptions will use NS vector base address given by this port.
initvtorns	slave	Value	Reset configuration port - Non-Secure Vector table offset (VTOR.TBLOFF[31:7]) out of reset It becomes functional when ARMv8-M Security Extensions are included When ARMv8-M Security Extensions are not included, this port will be ignored.
io_port_in	slave	PVBus	I/O port pair. See the documentation for the io_port_out port.
io_port_out	master	PVBus	I/O port pair. Used if IOP is true. Transactions from io_port_out which do not "match" should be returned via io_port_in. For performance reasons, the I/O port interface is not modelled directly. Instead, a simple PVBus gasket is inserted at the point in the memory system where the I/O port would be. In hardware, a device would be attached to the port which would tell the CPU whether it would like to intercept each transaction, given its address. This can be modelled in a performant manner by connecting a PVBusMapper-based device to io_port_out which intercepts transactions of interest and passes other transactions back to the CPU via io_port_in. Your I/O port device model is also responsible for aborting transactions which would be aborted on hardware (e.g. exclusives) if necessary.
irq	slave	Signal	This signal array delivers signals to the NVIC.
lockup	master	Signal	Asserted when the processor is in lockup state.

Port	Direction	Protocol	Description
niden	slave	Signal	Non-invasive debug enable, Set=enabled, Clear=disabled
nmi	slave	Signal	Configure non maskable interrupt.
poreset	slave	Signal	Raising this signal will do a power-on reset of the core.
pv_ppbus_m	master	PVBus	The core will generate External Private Peripheral Bus requests on this port.
pvbus_m	master	PVBus	The core will generate bus requests on this port.
sleepdeep	master	Signal	Asserted when the processor is in deep sleep.
sleeping	master	Signal	Asserted when the processor is in sleep.
spiden	slave	Signal	Secure Debug enable , Set=enabled, Clear=disabled
spniden	slave	Signal	Secure Non-invasive debug enable, Set=enabled, Clear=disabled
stcalib	slave	Value	This is the calibration value for the Secure (or only, when ARMv8-M Security Extensions are not included) SysTick timer. It is referred to as CFGSTCALIB in the M23 specification. The following bit flag is modelled stcalib[25] - NOREF - Indicates no reference clock integrated. If stclk is not in use, set to 1
stcalibns	slave	Value	This is the calibration value for the Non-Secure SysTick timer. When ARMv8-M Security Extensions are not included, this port will be ignored. It is referred to as CFGSTCALIBNS in the M23 specification. The following bit flag is modelled stcalibns[25] - NOREF - Indicates no reference clock integrated. If stclk is not in use, set to 1
stclk	slave	ClockSignal	This is the reference clock for the SysTick timer, called STCLKEN/STCLKENNS in the specification. This clock is provided for both secure and non secure inputs, even where ARMv8-M Security Extensions are included.
sysresetreq	master	Signal	Asserted to indicate that a reset is required.
ticks	master	InstructionCount	Port allowing the number of instructions since startup to be read from the CPU.
wicenack	master	Signal	Acknowledge signal for WICENREQ
wicenreq	slave	Signal	Request for deep sleep to be WIC-based deep sleep.
wicsense	master	Signal	Indicates which input events can cause the WIC to generate the WAKEUP signal.

## Parameters for ARM CortexM23CT

### SAU\_REGIONX.BADDR

Base address of SAU region0 at reset.

Type: `uint32_t`

Default value: `0x0`

### SAU\_REGIONX.ENABLE

Enable SAU region0 at reset.

Type: `bool`

Default value: `false`

**SAU\_REGIONX.LADDR**

Limit address of SAU region0 at reset.

Type: `uint32_t`

Default value: `0x0`

**SAU\_REGIONX.NSC**

Set NSC for SAU region0 at reset.

Type: `bool`

Default value: `false`

**BE**

Initialize processor to big endian mode.

Type: `bool`

Default value: `false`

**BKPT**

Number of breakpoint unit comparators implemented.

Type: `uint8_t`

Default value: `4`

**CTI**

CTI (Cross Trigger Interface) included.

Type: `bool`

Default value: `false`

**CTI\_irq0\_pin**

CTI interrupt request 0 pin.

Type: `uint8_t`

Default value: `4`

**CTI\_irq1\_pin**

CTI interrupt request 1 pin.

Type: `uint8_t`

Default value: `5`



**DBG**

Set whether debug extensions are implemented.

Type: `bool`

Default value: `true`

**INITVTOR**

Secure vector-table offset at reset.

Type: `uint32_t`

Default value: `0x0`

**INITVTORNS**

Non-Secure vector-table offset at reset.

Type: `uint32_t`

Default value: `0x0`

**IOP**

Send all d-side transactions to the port, `io_port_out`. Transactions which do not match should be returned to the port, `io_port_in`.

Type: `bool`

Default value: `false`

**IRQDIS0**

IRQ line disable mask. Bit `n` of this 32-bit parameter disables `IRQ[n+0]`.

Type: `uint32_t`

Default value: `0x0`

**IRQDIS1**

IRQ line disable mask. Bit `n` of this 32-bit parameter disables `IRQ[n+32]`.

Type: `uint32_t`

Default value: `0x0`

**IRQDIS2**

IRQ line disable mask. Bit `n` of this 32-bit parameter disables `IRQ[n+64]`.

Type: `uint32_t`

Default value: 0x0

**IRQDIS3**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+96].

Type: uint32\_t

Default value: 0x0

**IRQDIS4**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+128].

Type: uint32\_t

Default value: 0x0

**IRQDIS5**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+160].

Type: uint32\_t

Default value: 0x0

**IRQDIS6**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+192].

Type: uint32\_t

Default value: 0x0

**IRQDIS7**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+224].

Type: uint32\_t

Default value: 0x0

**MPU\_NS**

Number of regions in the Non-Secure MPU. If Security Extensions are absent, this is the total number of MPU regions.

Type: uint8\_t

Default value: 8

**MPU\_S**

Number of regions in the Secure MPU. If Security Extensions are absent, this is ignored.

Type: `uint8_t`

Default value: 8

### **NUMIRQ**

Number of user interrupts.

Type: `uint8_t`

Default value: 16

### **SAU**

Number of SAU regions (0 => no SAU).

Type: `uint8_t`

Default value: 4

### **SAU\_CTRL.ALLNS**

At reset, the SAU treats entire memory space as NS when the SAU is disabled if this is set.

Type: `bool`

Default value: `false`

### **SAU\_CTRL.ENABLE**

Enable SAU at reset.

Type: `bool`

Default value: `false`

### **SECEXT**

Whether the ARMv8-M Security Extensions are included.

Type: `bool`

Default value: `true`

### **SYST**

Include SysTick timer functionality (0=Absent, 1=Secure only, 2=Secure and NS).

Type: `uint8_t`

Default value: 2

### **VTOR**

Include Vector Table Offset Register.

Type: `bool`

Default value: `true`

### **WIC**

Include support for WIC-mode deep sleep.

Type: `bool`

Default value: `true`

### **WICLINES**

Number of lines supported by the WIC interface.

Type: `uint8_t`

Default value: 18

### **WPT**

Number of watchpoint unit comparators implemented.

Type: `uint8_t`

Default value: 4

### **cpi\_div**

divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: 0x1

### **cpi\_mul**

multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: 0x1

### **has\_core\_dside\_bus\_gasket**

STL gasket enabled.

Type: `bool`

Default value: `false`

### **manager\_id**

Manager ID presented in bus transactions.

Type: `uint64_t`

Default value: `0x0`

### **min\_sync\_level**

force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

### **reported\_patch\_level**

Purely cosmetic patch level value to be displayed from the CUID register. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: `int8_t`

Default value: `-1`

### **reported\_revision\_number**

Purely cosmetic revision number value to be displayed from the CUID register. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: `int8_t`

Default value: `-1`

### **semihosting-Thumb\_SVC**

T32 SVC number for semihosting.

Type: `uint8_t`

Default value: `171`

### **semihosting-cmd\_line**

Command line available to semihosting SVC calls.

Type: `string`

Default value: `N/A`

### **semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**semihosting-enable**

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

Type: bool  
Default value: true

**semihosting-heap\_base**

Virtual address of heap base.

Type: uint32\_t  
Default value: 0x0

**semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint32\_t  
Default value: 0x20700000

**semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint32\_t  
Default value: 0x20800000

**semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint32\_t  
Default value: 0x20700000

3.82 ARMCortexM33CT

Defined in LISA/ARMCortexM33CT.lisa.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r1p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

## Changes in 11.31.15

The following ports were removed:

- `auxfault`
- `currpri`

## Differences between the model and the RTL

The model does not support the following:

- ETM, MTB, or TPIU. MTB RAM is absent on the model.
- The power control (Q-Channel) interface.

## Implementation of ITM in M-class models

This model has a parameter that enables partial support for Instrumentation Trace Macrocell (ITM). In hardware or RTL, trace data from ITM is sent in packets to the trace block serially using a single pin or wire. In the model, if it is enabled, the ITM trace data is output using an MTI trace source called ITM. The ITM trace source has an `ITM_PACKET_TYPE` field. The following table shows which packet types the model supports:

Field value	Description	Supported by model
ITM_SYNC	Synchronization packet	Not supported.
ITM_P_OVERFLOW	Protocol: Overflow packet	Not supported.
ITM_P_LOCAL_TIMESTAMP	Protocol: Local timestamp packets	Not supported.
ITM_P_GLOBAL_TIMESTAMP	Protocol: Global timestamp packets	Not supported.
ITM_P_EXTEN	Protocol: Extension packet	Not supported.
ITM_S_INSTRUMENTATION	Source: Instrumentation packet	Supported.
ITM_S_DWT_EVENT_COUNTER	Hardware source: Event counter wrapping	Not supported.
ITM_S_DWT_EXCEPTION	Hardware source: Exception tracing	Supported.
ITM_S_DWT_PC_SAMPLING	Hardware source: PC sampling	Not supported.
ITM_S_DWT_DATA_PC_TRACE	Hardware source: DWT Data trace PC value	Supported.
ITM_S_DWT_DATA_ADDRESS_TRACE	Hardware source: DWT Data trace address value	Supported.
ITM_S_DWT_DATA_DATA_TRACE	Hardware source: DWT Data trace DATA value	Supported.

## Iris and MTI instances for ARM CortexM33CT

This model has the following Iris instances:

Name	Instance type
ARM CortexM33CT	<a href="#">ARM_Cortex-M33</a>
ARM CortexM33CT.acp_mapper	<a href="#">PVBUSMapper</a>
ARM CortexM33CT.ext_bus	<a href="#">PVBUSLogger</a>
ARM CortexM33CT.ext_bus.mapper	<a href="#">PVBUSMapper</a>
ARM CortexM33CT.l2_flusher	<a href="#">AsyncCacheFlushUnit</a>

This model has the following MTI trace components:

Name	Component type
ARMCortexM33CT	ARM_Cortex-M33
ARMCortexM33CT.acp_mapper	PVBusMapper
ARMCortexM33CT.ext_bus	PVBusLogger
ARMCortexM33CT.ext_bus.mapper	PVBusMapper
ARMCortexM33CT.l2_flusher	AsyncCacheFlushUnit

## Ports for ARMCortexM33CT

Port	Direction	Protocol	Description
ahbd	slave	PVBus	Debug AHB - core bus slave driven by the DAP.
bigend	slave	Signal	Configure big endian data format.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions. Referred to in some documents as 'CLKIN'.
coproc_bus	slave	CoprocBusProtocol	Co-Processor Interface
cpuwait	slave	Signal	Stall the CPU out of reset
currns	master	Signal	Current Security state of the processor
dbgen	slave	Signal	Invasive debug enable
dbgrestart	slave	Signal	Request for synchronised exit from halt mode
dbgrestarted	master	Signal	Handshakes with DBGRESTART
edbgrq	slave	Signal	External request to enter halt mode
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE and corresponds to the RTL TXEV and RXEV signals.
fpxxc	master	Value	Port that sends the value of the FPxxC exception flags (FPIX, FPIDC, FPOFC, FPUFC, FPDZC, FPIOC). Each flag is present in the same position as in the FPSCR register (e.g. fpxxc[0] = FPIOC).
halted	master	Signal	Indicates that the processor is in halt mode
idau_invalidate_region	slave	Value_64	64 bit number to invalid IDAU memory range (start_address<<32 end_address)
idau	master	PVBus	The core will generate IDAU requests on this port.
initnsvtor	slave	Value	Reset configuration port - Non-Secure Vector table offset (VTOR.TBLOFF[31:7]) out of reset This port remains functional no matter ARMv8-M Security Extensions are included or not When ARMv8-M Security Extensions are not included, all exceptions will use NS vector base address given by this port.
initsvtor	slave	Value	Reset configuration port - Secure Vector table offset (VTOR.TBLOFF[31:7]) out of reset It becomes functional when ARMv8-M Security Extensions are included When ARMv8-M Security Extensions are not included, this port will be ignored.
intnum	master	Value	Exception number of the current execution context (from IPSR[8:0]) When the processor is in Thread mode, INTNUM is 0 When the processor is in Handler mode, INTNUM is the exception number of the currently-executing exception.
irq	slave	Signal	This signal array delivers signals to the NVIC.
locknsmpu	slave	Signal	Disable writes to the Non-Secure MPU_*_NS registers
locknsvtor	slave	Signal	Cortex-M33-specific LOCKNSVTOR, LOCKSVTAIRCR, LOCKSMPU, LOCKNSMPU, LOCKSAU. Disable writes to VTOR_NS
locksau	slave	Signal	Disable writes to the SAU_* registers
locksmpu	slave	Signal	Disable writes to the Secure MPU_* registers



Port	Direction	Protocol	Description
locksvtaircr	slave	Signal	Disable writes to VTOR_S, AIRCR.PRIS, AIRCR.BFHFNMINS
lockup	master	Signal	Asserted when the processor is in lockup state.
niden	slave	Signal	Non-invasive debug enable
nmi	slave	Signal	Configure non maskable interrupt.
poreset	slave	Signal	Raising this signal will do a power-on reset of the core.
pv_ppbus_m	master	PVBus	The core will generate External Private Peripheral Bus requests on this port.
pvbus_m	master	PVBus	The core will generate bus requests on this port.
sleepdeep	master	Signal	Asserted when the processor is in deep sleep.
sleeping	master	Signal	Asserted when the processor is in sleep.
spiden	slave	Signal	Secure invasive debug enable
spniden	slave	Signal	Secure non-invasive debug enable
stcalib	slave	Value	These are the secure and non-secure calibration values for the SysTick timer, called CFGSSTCALIB/CFGNSSTCALIB in the M33 specification. stcalib[0] is the secure timer calibration, stcalib[1] is the non-secure timer calibration. In the case where the Security Extension is not present, stcalib[0] should be used. The following bit flag is modelled for each stcalib[*][25] - NOREF - Indicates no reference clock integrated. If stclk is not in use, set to 1
stclk	slave	ClockSignal	This is the reference clock for the SysTick timer, called SSTCLKEN/NSSTCLKEN in the M33 specification. This clock is provided for both secure and non secure inputs, even where ARMv8-M Security Extensions are included.
sysreset	slave	Signal	Raising this signal will put the core into reset mode (but does not reset the debug logic).
sysresetreq	master	Signal	Asserted to indicate that a reset is required.
ticks	master	InstructionCount	Port allowing the number of instructions since startup to be read from the CPU.
wicenack	master	Signal	Acknowledge signal for WICENREQ
wicenreq	slave	Signal	Request for deep sleep to be WIC-based deep sleep.
wicsense	master	Signal	Indicates which input events can cause the WIC to generate the WAKEUP signal.

## Parameters for ARM CortexM33CT

### SAU\_REGIONX.BADDR

Base address of SAU region0 at reset.

Type: `uint32_t`

Default value: `0x0`

### SAU\_REGIONX.ENABLE

Enable SAU region0 at reset.

Type: `bool`

Default value: `false`

**SAU\_REGIONX.LADDR**

Limit address of SAU region0 at reset.

Type: `uint32_t`

Default value: `0x0`

**SAU\_REGIONX.NSC**

Set NSC for SAU region0 at reset.

Type: `bool`

Default value: `false`

**BIGENDINIT**

Initialize processor to big endian mode.

Type: `bool`

Default value: `false`

**CDEMAPPEDONCP**

Bit N specifies whether the instruction for coprocessor N (CP7:CP0) is redirected to the CDE module.

Type: `uint8_t`

Default value: `255`

**CDERTLID**

Value of ID\_AFR0.CDERTLID.

Type: `uint8_t`

Default value: `32`

**CFGNOCDECP**

Bit N means external coprocessor N (CP7:CP0) disable for CDE coprocessor.

Type: `uint8_t`

Default value: `0`

**CPIF**

Specifies whether the external coprocessor interface is included.

Type: `bool`

Default value: `true`

**CPNSPRESENT**

Bit N means external coprocessor N (CP7:CP0) is accessible in Non-Secure state.

Type: `uint8_t`

Default value: 255

**CPSPPRESENT**

Bit N means external coprocessor N (CP7:CP0) is accessible in Secure state.

Type: `uint8_t`

Default value: 255

**CTI**

CTI (Cross Trigger Interface) included.

Type: `bool`

Default value: `false`

**CTI\_irq0\_pin**

CTI interrupt request 0 pin.

Type: `uint16_t`

Default value: 4

**CTI\_irq1\_pin**

CTI interrupt request 1 pin.

Type: `uint16_t`

Default value: 5

**DBGLVL**

0: Minimal debug; 1: 2 Watchpoints, 4 Breakpoint comparators; 2: 4 Watchpoints, 8 Breakpoint comparators.

Type: `uint8_t`

Default value: 2

**DSP**

Set whether the model has the DSP extension.

Type: `bool`

Default value: `true`

### **FPU**

Set whether the model has VFP support.

Type: `bool`

Default value: `true`

### **INITNSVTOR**

Non-Secure vector-table offset at reset.

Type: `uint32_t`

Default value: `0x0`

### **INITSVTOR**

Secure vector-table offset at reset.

Type: `uint32_t`

Default value: `0x0`

### **IRQDIS0**

IRQ line disable mask. Bit *n* of this 32-bit parameter disables IRQ[*n*+0].

Type: `uint32_t`

Default value: `0x0`

### **IRQDIS1**

IRQ line disable mask. Bit *n* of this 32-bit parameter disables IRQ[*n*+32].

Type: `uint32_t`

Default value: `0x0`

### **IRQDIS10**

IRQ line disable mask. Bit *n* of this 32-bit parameter disables IRQ[*n*+320].

Type: `uint32_t`

Default value: `0x0`

### **IRQDIS11**

IRQ line disable mask. Bit *n* of this 32-bit parameter disables IRQ[*n*+352].

Type: `uint32_t`

Default value: `0x0`

### **IRQDIS12**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+384].

Type: `uint32_t`

Default value: `0x0`

### **IRQDIS13**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+416].

Type: `uint32_t`

Default value: `0x0`

### **IRQDIS14**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+448].

Type: `uint32_t`

Default value: `0x0`

### **IRQDIS2**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+64].

Type: `uint32_t`

Default value: `0x0`

### **IRQDIS3**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+96].

Type: `uint32_t`

Default value: `0x0`

### **IRQDIS4**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+128].

Type: `uint32_t`

Default value: `0x0`

### **IRQDIS5**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+160].

Type: `uint32_t`

Default value: `0x0`

### **IRQDIS6**

IRQ line disable mask. Bit *n* of this 32-bit parameter disables IRQ[*n*+192].

Type: `uint32_t`

Default value: `0x0`

### **IRQDIS7**

IRQ line disable mask. Bit *n* of this 32-bit parameter disables IRQ[*n*+224].

Type: `uint32_t`

Default value: `0x0`

### **IRQDIS8**

IRQ line disable mask. Bit *n* of this 32-bit parameter disables IRQ[*n*+256].

Type: `uint32_t`

Default value: `0x0`

### **IRQDIS9**

IRQ line disable mask. Bit *n* of this 32-bit parameter disables IRQ[*n*+288].

Type: `uint32_t`

Default value: `0x0`

### **IRQLVL**

Number of bits of interrupt priority.

Type: `uint8_t`

Default value: `3`

### **ITM**

Level of instrumentation trace supported. `false` : No ITM trace included, `true`: ITM trace included.

Type: `bool`

Default value: `true`

### **LOCK\_NS\_MPU**

Lock down of Non-Secure MPU registers write.

Type: `bool`

Default value: `false`

### **LOCK\_SAU**

Lock down of SAU registers write.

Type: `bool`

Default value: `false`

### **LOCK\_S\_MPU**

Lock down of Secure MPU registers write.

Type: `bool`

Default value: `false`

### **MPU\_NS**

Number of regions in the Non-Secure MPU. If Security Extensions are absent, this is the total number of MPU regions.

Type: `uint8_t`

Default value: 8

### **MPU\_S**

Number of regions in the Secure MPU. If Security Extensions are absent, this is ignored.

Type: `uint8_t`

Default value: 8

### **NUMIRQ**

Number of user interrupts.

Type: `uint16_t`

Default value: 32

### **SAU**

Number of SAU regions (0 => no SAU).

Type: `uint8_t`

Default value: 4

**SAU\_CTRL.ALLNS**

At reset, the SAU treats entire memory space as NS when the SAU is disabled if this is set.

Type: `bool`

Default value: `false`

**SAU\_CTRL.ENABLE**

Enable SAU at reset.

Type: `bool`

Default value: `false`

**SECEXT**

Whether the ARMv8-M Security Extensions are included.

Type: `bool`

Default value: `true`

**WIC**

Include support for WIC-mode deep sleep.

Type: `bool`

Default value: `true`

**WICLINES**

Number of lines supported by the WIC interface.

Type: `uint16_t`

Default value: 35

**cde\_impl\_name**

Name of the CDE implementation for this core (implementation contributed by MTI plugin).

Type: `string`

Default value: N/A

**cpi\_div**

divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: 0x1



**cpi\_mul**

multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**has\_cde**

Enables Custom Datapath Extensions.

Type: `bool`

Default value: `false`

**manager\_id**

Manager ID presented in bus transactions.

Type: `uint64_t`

Default value: `0x0`

**min\_sync\_level**

force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

**reported\_patch\_level**

Purely cosmetic patch level value to be displayed from the CPUID register. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: `int8_t`

Default value: `-1`

**reported\_revision\_number**

Purely cosmetic revision number value to be displayed from the CPUID register. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: `int8_t`

Default value: `-1`

**semihosting-Thumb\_SVC**

T32 SVC number for semihosting.

Type: `uint8_t`

Default value: 171

### **semihosting-cmd\_line**

Command line available to semihosting SVC calls.

Type: `string`

Default value: N/A

### **semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **semihosting-enable**

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

Type: `bool`

Default value: `true`

### **semihosting-heap\_base**

Virtual address of heap base.

Type: `uint32_t`

Default value: 0x0

### **semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint32_t`

Default value: 0x20700000

### **semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint32_t`

Default value: 0x20800000

**semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint32_t`

Default value: `0x20700000`

**vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

## 3.83 ARMCortexM35PCT

Defined in `LISA/ARMCortexM35PCT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r1p2	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following ports were removed:

- `auxfault`
- `currpri`

### Differences between the model and the RTL

- The model does not support the following:
  - ETM, MTB, or TPIU. MTB RAM is absent on the model.
  - Caches.
  - The co-processor interface.
  - The power control (Q-Channel) interface.
- The model does not implement any physical security features.
- Bits[3:0] of the Anti-tampering Features Control Register are supported for read/write. No functionality is implemented.
- Read/write access to the Anti-tampering Features Control Register is supported using SECKEY. No functionality is implemented.

## Implementation of ITM in M-class models

This model has a parameter that enables partial support for Instrumentation Trace Macrocell (ITM). In hardware or RTL, trace data from ITM is sent in packets to the trace block serially using a single pin or wire. In the model, if it is enabled, the ITM trace data is output using an MTI trace source called ITM. The ITM trace source has an `ITM_PACKET_TYPE` field. The following table shows which packet types the model supports:

Field value	Description	Supported by model
ITM_SYNC	Synchronization packet	Not supported.
ITM_P_OVERFLOW	Protocol: Overflow packet	Not supported.
ITM_P_LOCAL_TIMESTAMP	Protocol: Local timestamp packets	Not supported.
ITM_P_GLOBAL_TIMESTAMP	Protocol: Global timestamp packets	Not supported.
ITM_P_EXTEN	Protocol: Extension packet	Not supported.
ITM_S_INSTRUMENTATION	Source: Instrumentation packet	Supported.
ITM_S_DWT_EVENT_COUNTER	Hardware source: Event counter wrapping	Not supported.
ITM_S_DWT_EXCEPTION	Hardware source: Exception tracing	Supported.
ITM_S_DWT_PC_SAMPLING	Hardware source: PC sampling	Not supported.
ITM_S_DWT_DATA_PC_TRACE	Hardware source: DWT Data trace PC value	Supported.
ITM_S_DWT_DATA_ADDRESS_TRACE	Hardware source: DWT Data trace address value	Supported.
ITM_S_DWT_DATA_DATA_TRACE	Hardware source: DWT Data trace DATA value	Supported.

## Iris and MTI instances for ARM CortexM35PCT

This model has the following Iris instances:

Name	Instance type
ARM CortexM35PCT	ARM_Cortex-M35P
ARM CortexM35PCT.acp_mapper	PVBusMapper
ARM CortexM35PCT.ext_bus	PVBusLogger
ARM CortexM35PCT.ext_bus.mapper	PVBusMapper
ARM CortexM35PCT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

Name	Component type
ARM CortexM35PCT	ARM_Cortex-M35P
ARM CortexM35PCT.acp_mapper	PVBusMapper
ARM CortexM35PCT.ext_bus	PVBusLogger
ARM CortexM35PCT.ext_bus.mapper	PVBusMapper
ARM CortexM35PCT.l2_flusher	AsyncCacheFlushUnit

## Ports for ARM CortexM35PCT

Port	Direction	Protocol	Description
bigend	slave	Signal	Configure big endian data format.

Port	Direction	Protocol	Description
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions. Referred to in some documents as 'CLKIN'.
coproc_bus	slave	CoprocBusProtocol	Co-Processor Interface
cpuwait	slave	Signal	Stall the CPU out of reset
currns	master	Signal	Current Security state of the processor
dbgen	slave	Signal	Invasive debug enable
dbgrestart	slave	Signal	Request for synchronised exit from halt mode
dbgrestarted	master	Signal	Handshakes with DBGRESTART
edbgrq	slave	Signal	External request to enter halt mode
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE and corresponds to the RTL TXEV and RXEV signals.
fpxxc	master	Value	Port that sends the value of the FPxxC exception flags (FPIX, FPIDC, FPOFC, FPUFC, FPDZC, FPIOC). Each flag is present in the same position as in the FPSCR register (e.g. fpxxc[0] = FPIOC).
halted	master	Signal	Indicates that the processor is in halt mode
idau_invalidate_region	slave	Value_64	64 bit number to invalid IDAU memory range (start_address<<32 end_address)
idau	master	PVBus	The core will generate IDAU requests on this port.
initnsvtor	slave	Value	Reset configuration port - Non-Secure Vector table offset (VTOR.TBLOFF[31:7]) out of reset This port remains functional no matter ARMv8-M Security Extensions are included or not When ARMv8-M Security Extensions are not included, all exceptions will use NS vector base address given by this port.
initsvtor	slave	Value	Reset configuration port - Secure Vector table offset (VTOR.TBLOFF[31:7]) out of reset It becomes functional when ARMv8-M Security Extensions are included When ARMv8-M Security Extensions are not included, this port will be ignored.
intnum	master	Value	Exception number of the current execution context (from IPSR[8:0]) When the processor is in Thread mode, INTNUM is 0 When the processor is in Handler mode, INTNUM is the exception number of the currently-executing exception.
irq	slave	Signal	This signal array delivers signals to the NVIC.
LOCKATFCR	slave	Signal	Port Lock ATFCR register
locknsmpu	slave	Signal	Disable writes to the Non-Secure MPU_*_NS registers
locknsvtor	slave	Signal	Disable writes to VTOR_NS
locksau	slave	Signal	Disable writes to the SAU_* registers
locksmmu	slave	Signal	Disable writes to the Secure MPU_* registers
locksvtaircr	slave	Signal	Disable writes to VTOR_S, AIRCR.PRIS, AIRCR.BFHFNMINS
lockup	master	Signal	Asserted when the processor is in lockup state.
niden	slave	Signal	Non-invasive debug enable
nmi	slave	Signal	Configure non maskable interrupt.
poreset	slave	Signal	Raising this signal will do a power-on reset of the core.
pv_ppbus_m	master	PVBus	The core will generate External Private Peripheral Bus requests on this port.
pdbus_m	master	PVBus	The core will generate bus requests on this port.
sleepdeep	master	Signal	Asserted when the processor is in deep sleep.
sleeping	master	Signal	Asserted when the processor is in sleep.
spiden	slave	Signal	Secure invasive debug enable

Port	Direction	Protocol	Description
spniden	slave	Signal	Secure non-invasive debug enable
stcalib	slave	Value	These are the secure and non-secure calibration values for the SysTick timer. stcalib[0] is the secure timer calibration, stcalib[1] is the non-secure timer calibration. In the case where the Security Extension is not present, stcalib[0] should be used. The following bit flag is modelled for each stcalib[*][25] - NOREF - Indicates no reference clock integrated. If stclk is not in use, set to 1
stclk	slave	ClockSignal	This is the reference clock for the SysTick timer, called SSTCLKEN/NSSTCLKEN in the M33 specification. This clock is provided for both secure and non secure inputs, even where ARMv8-M Security Extensions are included.
sysreset	slave	Signal	Raising this signal will put the core into reset mode (but does not reset the debug logic).
sysresetreq	master	Signal	Asserted to indicate that a reset is required.
ticks	master	InstructionCount	Port allowing the number of instructions since startup to be read from the CPU.
wicenack	master	Signal	Acknowledge signal for WICENREQ
wicenreq	slave	Signal	Request for deep sleep to be WIC-based deep sleep.
wicsense	master	Signal	Indicates which input events can cause the WIC to generate the WAKEUP signal.

## Parameters for ARM Cortex M35PCT

### SAU\_REGIONX.BADDR

Base address of SAU region0 at reset.

Type: `uint32_t`

Default value: `0x0`

### SAU\_REGIONX.ENABLE

Enable SAU region0 at reset.

Type: `bool`

Default value: `false`

### SAU\_REGIONX.LADDR

Limit address of SAU region0 at reset.

Type: `uint32_t`

Default value: `0x0`

### SAU\_REGIONX.NSC

Set NSC for SAU region0 at reset.

Type: `bool`

Default value: `false`

**ATFINITEN**

ATFCR is enabled when the core goes out of reset.

Type: `bool`

Default value: `false`

**BIGENDINIT**

Initialize processor to big endian mode.

Type: `bool`

Default value: `false`

**CPIF**

Specifies whether the external coprocessor interface is included.

Type: `bool`

Default value: `true`

**CPNSPRESENT**

Bit N means external coprocessor N (CP7:CP0) is accessible in Non-Secure state.

Type: `uint8_t`

Default value: 255

**CPSPRESENT**

Bit N means external coprocessor N (CP7:CP0) is accessible in Secure state.

Type: `uint8_t`

Default value: 255

**CTI**

CTI (Cross Trigger Interface) included.

Type: `bool`

Default value: `false`

**CTI\_irq0\_pin**

CTI interrupt request 0 pin.

Type: `uint16_t`

Default value: 4

**CTI\_irq1\_pin**

CTI interrupt request 1 pin.

Type: `uint16_t`

Default value: 5

**DBGLVL**

0: Minimal debug; 1: 2 Watchpoints, 4 Breakpoint comparators; 2: 4 Watchpoints, 8 Breakpoint comparators.

Type: `uint8_t`

Default value: 2

**DSP**

Set whether the model has the DSP extension.

Type: `bool`

Default value: `true`

**FPU**

Set whether the model has VFP support.

Type: `bool`

Default value: `true`

**INITNSVTOR**

Non-Secure vector-table offset at reset.

Type: `uint32_t`

Default value: `0x0`

**INITSVTOR**

Secure vector-table offset at reset.

Type: `uint32_t`

Default value: `0x0`

**IRQDIS0**

IRQ line disable mask. Bit *n* of this 32-bit parameter disables `IRQ[n+0]`.

Type: `uint32_t`



Default value: 0x0

**IRQDIS1**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+32].

Type: uint32\_t

Default value: 0x0

**IRQDIS10**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+320].

Type: uint32\_t

Default value: 0x0

**IRQDIS11**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+352].

Type: uint32\_t

Default value: 0x0

**IRQDIS12**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+384].

Type: uint32\_t

Default value: 0x0

**IRQDIS13**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+416].

Type: uint32\_t

Default value: 0x0

**IRQDIS14**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+448].

Type: uint32\_t

Default value: 0x0

**IRQDIS2**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+64].

Type: uint32\_t

Default value: 0x0

**IRQDIS3**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+96].

Type: uint32\_t

Default value: 0x0

**IRQDIS4**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+128].

Type: uint32\_t

Default value: 0x0

**IRQDIS5**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+160].

Type: uint32\_t

Default value: 0x0

**IRQDIS6**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+192].

Type: uint32\_t

Default value: 0x0

**IRQDIS7**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+224].

Type: uint32\_t

Default value: 0x0

**IRQDIS8**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+256].

Type: uint32\_t

Default value: 0x0

**IRQDIS9**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+288].

Type: uint32\_t

Default value: 0x0

**IRQLVL**

Number of bits of interrupt priority.

Type: `uint8_t`

Default value: 3

**ITM**

Level of instrumentation trace supported. `false` : No ITM trace included, `true`: ITM trace included.

Type: `bool`

Default value: `true`

**LOCK\_NS\_MPU**

Lock down of Non-Secure MPU registers write.

Type: `bool`

Default value: `false`

**LOCK\_SAU**

Lock down of SAU registers write.

Type: `bool`

Default value: `false`

**LOCK\_S\_MPU**

Lock down of Secure MPU registers write.

Type: `bool`

Default value: `false`

**MPU\_NS**

Number of regions in the Non-Secure MPU. If Security Extensions are absent, this is the total number of MPU regions.

Type: `uint8_t`

Default value: 8

**MPU\_S**

Number of regions in the Secure MPU. If Security Extensions are absent, this is ignored.

Type: `uint8_t`

Default value: 8

### **NUMIRQ**

Number of user interrupts.

Type: `uint16_t`

Default value: 32

### **SAU**

Number of SAU regions (0 => no SAU).

Type: `uint8_t`

Default value: 4

### **SAU\_CTRL.ALLNS**

At reset, the SAU treats entire memory space as NS when the SAU is disabled if this is set.

Type: `bool`

Default value: `false`

### **SAU\_CTRL.ENABLE**

Enable SAU at reset.

Type: `bool`

Default value: `false`

### **SECEXT**

Whether the ARMv8-M Security Extensions are included.

Type: `bool`

Default value: `true`

### **WIC**

Include support for WIC-mode deep sleep.

Type: `bool`

Default value: `true`

### **WICLINES**

Number of lines supported by the WIC interface.

Type: `uint16_t`

Default value: 35

### **`cpi_div`**

divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: 0x1

### **`cpi_mul`**

multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: 0x1

### **`manager_id`**

Manager ID presented in bus transactions.

Type: `uint64_t`

Default value: 0x0

### **`min_sync_level`**

force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: 0

### **`reported_patch_level`**

Purely cosmetic patch level value to be displayed from the CPUID register. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: `int8_t`

Default value: -1

### **`reported_revision_number`**

Purely cosmetic revision number value to be displayed from the CPUID register. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: `int8_t`

Default value: -1

**semihosting-Thumb\_SVC**

T32 SVC number for semihosting.

Type: `uint8_t`

Default value: 171

**semihosting-cmd\_line**

Command line available to semihosting SVC calls.

Type: `string`

Default value: N/A

**semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**semihosting-enable**

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

Type: `bool`

Default value: `true`

**semihosting-heap\_base**

Virtual address of heap base.

Type: `uint32_t`

Default value: `0x0`

**semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint32_t`

Default value: `0x20700000`

**semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint32_t`

Default value: 0x20800000

### **semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint32\_t

Default value: 0x20700000

### **vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

## 3.84 ARMCortexM3CT

Defined in LISA/ARMCortexM3CT.lisa.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r2p1	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Differences between the model and the RTL

This component has the following differences from the corresponding revision of the RTL implementation:

- The WIC is not currently implemented.
- Power control is not implemented, so the processor does not set the SLEEPING or SLEEPDEEP signals. It does not support powering down of the processor.
- Only the minimal level of debug support is provided (no DAP, FPB, DWT, or halting debug capability).
- Debug-related components are not implemented.
- The unimplemented registers are the processor debug registers, system debug registers, debug interface port registers, TPIU registers, and ETM registers.
- The processor must still be clocked even if it has asserted the sleeping or sleepdeep signals.
- Disabling processor features using the Auxiliary Control Register is not supported.
- Only a single pvbus\_m master port is provided. This combines the ICode, DCode, and System bus interfaces of the RTL. The external PPB bus is provided by the pv\_ppbus\_m master port.

- In privileged mode, STRT and LDRT to the PPB region are not forbidden access.
- No support for ETM, TPIU, or HTM.
- There is no supported equivalent of the RESET\_ALL\_REGS configuration setting in RTL (that forces all registers to have a well-defined value on reset).
- The RTL implements the ROM table as an external component on the External Private Peripheral Bus. In the CT model, the ROM table is implemented internally as a fallback if an external PPB access in the ROM table address region aborts. This permits the default ROM table to be overridden (by implementing an external component connected to the external PPB to handle accesses to these addresses) without requiring every user of the processor to implement and connect a ROM table component.

### Implementation of ITM in M-class models

This model has a parameter that enables partial support for Instrumentation Trace Macrocell (ITM). In hardware or RTL, trace data from ITM is sent in packets to the trace block serially using a single pin or wire. In the model, if it is enabled, the ITM trace data is output using an MTI trace source called ITM. The ITM trace source has an `ITM_PACKET_TYPE` field. The following table shows which packet types the model supports:

Field value	Description	Supported by model
ITM_SYNC	Synchronization packet	Not supported.
ITM_P_OVERFLOW	Protocol: Overflow packet	Not supported.
ITM_P_LOCAL_TIMESTAMP	Protocol: Local timestamp packets	Not supported.
ITM_P_GLOBAL_TIMESTAMP	Protocol: Global timestamp packets	Not supported.
ITM_P_EXTEN	Protocol: Extension packet	Not supported.
ITM_S_INSTRUMENTATION	Source: Instrumentation packet	Supported.
ITM_S_DWT_EVENT_COUNTER	Hardware source: Event counter wrapping	Not supported.
ITM_S_DWT_EXCEPTION	Hardware source: Exception tracing	Supported.
ITM_S_DWT_PC_SAMPLING	Hardware source: PC sampling	Not supported.
ITM_S_DWT_DATA_PC_TRACE	Hardware source: DWT Data trace PC value	Supported.
ITM_S_DWT_DATA_ADDRESS_TRACE	Hardware source: DWT Data trace address value	Supported.
ITM_S_DWT_DATA_DATA_TRACE	Hardware source: DWT Data trace DATA value	Supported.

### Iris and MTI instances for ARM CortexM3CT

This model has the following Iris instances:

Name	Instance type
ARM CortexM3CT	ARM_Cortex-M3
ARM CortexM3CT.acp_mapper	PVBusMapper
ARM CortexM3CT.ext_bus	PVBusLogger
ARM CortexM3CT.ext_bus.mapper	PVBusMapper
ARM CortexM3CT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:



Name	Component type
ARMCortexM3CT	ARM_Cortex-M3
ARMCortexM3CT.acp_mapper	PVBusMapper
ARMCortexM3CT.ext_bus	PVBusLogger
ARMCortexM3CT.ext_bus.mapper	PVBusMapper
ARMCortexM3CT.l2_flusher	AsyncCacheFlushUnit

## Ports for ARMCortexM3CT

Port	Direction	Protocol	Description
ahb_ap	slave	PVBus	Debug AHB - core bus slave driven by the DAP.
auxfault	slave	Value	This is wired to the Auxiliary Fault Status Register.
bigend	slave	Signal	Configure big endian data format.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
currpri	master	Value	Current execution priority.
edbgrq	slave	Signal	External debug request.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE and corresponds to the RTL TXEV and RXEV signals.
intisr	slave	Signal	This signal array delivers signals to the NVIC.
intnmi	slave	Signal	Configure non maskable interrupt.
lockup	master	Signal	Asserted when the processor is in lockup state.
poreset	slave	Signal	Raising this signal will do a power-on reset of the core.
pv_ppbus_m	master	PVBus	The core will generate External Private Peripheral Bus requests on this port.
pvbush_m	master	PVBus	The core will generate bus requests on this port.
sleepdeep	master	Signal	Asserted when the processor is in deep sleep.
sleeping	master	Signal	Asserted when the processor is in sleep.
stcalib	slave	Value	This is the calibration value for the SysTick timer The following bit flag is modelled stcalib[25] - NOREF - Indicates no reference clock integrated. If stclk is not in use, set to 1
stclk	slave	ClockSignal	This is the reference clock for the SysTick timer.
sysreset	slave	Signal	Raising this signal will put the core into reset mode (but does not reset the debug logic).
sysresetreq	master	Signal	Asserted to indicate that a reset is required.
ticks	master	InstructionCount	Port allowing the number of instructions since startup to be read from the CPU.

## Parameters for ARMCortexM3CT

### BB\_PRESENT

Enable bitbanding.

Type: bool

Default value: true

**BIGENDINIT**

Initialize processor to big endian mode.

Type: `bool`

Default value: `false`

**DBGLVL**

0: No debug; 1: Minimal debug; 2: Full debug without DWT address matching; 3: Full debug support with DWT data-comparators.

Type: `uint8_t`

Default value: 3

**LVL\_WIDTH**

Number of bits of interrupt priority.

Type: `uint8_t`

Default value: 3

**NUM\_IRQ**

Number of user interrupts.

Type: `uint8_t`

Default value: 16

**NUM\_MPU\_REGION**

Number of MPU regions.

Type: `uint8_t`

Default value: 8

**TRACE\_LVL**

Level of instrumentation trace supported. 0: No trace, 1: Standard trace. ITM, TPIU and DWT triggers and counters present, 2: Full trace. ITM, TPIU, ETM and DWT triggers and counters present.

Type: `uint8_t`

Default value: 1

**WIC**

Include support for WIC-mode deep sleep.

Type: `bool`

Default value: `true`

### **`cpi_div`**

divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

### **`cpi_mul`**

multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

### **`manager_id`**

Manager ID presented in bus transactions.

Type: `uint64_t`

Default value: `0x0`

### **`min_sync_level`**

force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

### **`reported_patch_level`**

Purely cosmetic patch level value to be displayed from the CPUID register. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: `int8_t`

Default value: `-1`

### **`reported_revision_number`**

Purely cosmetic revision number value to be displayed from the CPUID register. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: `int8_t`

Default value: `-1`

**semihosting-Thumb\_SVC**

T32 SVC number for semihosting.

Type: `uint8_t`

Default value: 171

**semihosting-cmd\_line**

Command line available to semihosting SVC calls.

Type: `string`

Default value: N/A

**semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**semihosting-enable**

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

Type: `bool`

Default value: `true`

**semihosting-heap\_base**

Virtual address of heap base.

Type: `uint32_t`

Default value: `0x0`

**semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint32_t`

Default value: `0x20700000`

**semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint32_t`

Default value: 0x20800000

### **semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint32\_t

Default value: 0x20700000

## 3.85 ARMCortexM4CT

Defined in `LISA/ARMCortexM4CT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p1	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Differences between the model and the RTL

This component has the following differences from the corresponding revision of the RTL implementation:

- The Wakeup Interrupt Controller (WIC) is not implemented.
- Power control is not implemented. Powering down of the processor is not supported. The processor must still be clocked even if it has asserted the sleeping or sleepdeep signals.
- Only the minimal level of debug support is provided (no DAP, FPB, DWT, or halting debug capability).
- No debug-related components are implemented.
- The unimplemented registers are the processor debug registers, system debug registers, debug interface port registers, TPIU registers, and ETM registers.
- No support for ETM, TPIU, or HTM.
- There is no supported equivalent of the RESET\_ALL\_REGS configuration setting in RTL (that forces all registers to have a well-defined value on reset).
- Disabling processor features using the Auxiliary Control Register is not supported.
- Only a single pvbus\_m master port is provided. This combines the ICode, DCode, and System bus interfaces of the RTL. The external PPB bus is provided by the pv\_ppbus\_m master port.
- In privileged mode, STRT and LDRT to the PPB region are not forbidden access.
- The RTL implements the ROM table as an external component on the External Private Peripheral Bus. In the CT model, the ROM table is implemented internally as a fallback if an external PPB access in the ROM table address region aborts. This permits the default ROM table to be overridden (by implementing an external component connected to the external

PPB to handle accesses to these addresses) without requiring every user of the processor to implement and connect a ROM table component.

- Because the CT model does not provide a DAP port or halting debug capability, the dbgen signal is ignored.

### Implementation of ITM in M-class models

This model has a parameter that enables partial support for Instrumentation Trace Macrocell (ITM). In hardware or RTL, trace data from ITM is sent in packets to the trace block serially using a single pin or wire. In the model, if it is enabled, the ITM trace data is output using an MTI trace source called ITM. The ITM trace source has an `ITM_PACKET_TYPE` field. The following table shows which packet types the model supports:

Field value	Description	Supported by model
ITM_SYNC	Synchronization packet	Not supported.
ITM_P_OVERFLOW	Protocol: Overflow packet	Not supported.
ITM_P_LOCAL_TIMESTAMP	Protocol: Local timestamp packets	Not supported.
ITM_P_GLOBAL_TIMESTAMP	Protocol: Global timestamp packets	Not supported.
ITM_P_EXTEN	Protocol: Extension packet	Not supported.
ITM_S_INSTRUMENTATION	Source: Instrumentation packet	Supported.
ITM_S_DWT_EVENT_COUNTER	Hardware source: Event counter wrapping	Not supported.
ITM_S_DWT_EXCEPTION	Hardware source: Exception tracing	Supported.
ITM_S_DWT_PC_SAMPLING	Hardware source: PC sampling	Not supported.
ITM_S_DWT_DATA_PC_TRACE	Hardware source: DWT Data trace PC value	Supported.
ITM_S_DWT_DATA_ADDRESS_TRACE	Hardware source: DWT Data trace address value	Supported.
ITM_S_DWT_DATA_DATA_TRACE	Hardware source: DWT Data trace DATA value	Supported.

### Iris and MTI instances for ARM CortexM4CT

This model has the following Iris instances:

Name	Instance type
ARM CortexM4CT	<a href="#">ARM_Cortex-M4</a>
ARM CortexM4CT.acp_mapper	<a href="#">PVBUSMapper</a>
ARM CortexM4CT.ext_bus	<a href="#">PVBUSLogger</a>
ARM CortexM4CT.ext_bus.mapper	<a href="#">PVBUSMapper</a>
ARM CortexM4CT.l2_flusher	<a href="#">AsyncCacheFlushUnit</a>

This model has the following MTI trace components:

Name	Component type
ARM CortexM4CT	<a href="#">ARM_Cortex-M4</a>
ARM CortexM4CT.acp_mapper	<a href="#">PVBUSMapper</a>
ARM CortexM4CT.ext_bus	<a href="#">PVBUSLogger</a>
ARM CortexM4CT.ext_bus.mapper	<a href="#">PVBUSMapper</a>
ARM CortexM4CT.l2_flusher	<a href="#">AsyncCacheFlushUnit</a>

## Ports for ARMCortexM4CT

Port	Direction	Protocol	Description
ahb_ap	slave	PVBus	Debug AHB - core bus slave driven by the DAP.
auxfault	slave	Value	This is wired to the Auxiliary Fault Status Register.
bigend	slave	Signal	Configure big endian data format.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
currpri	master	Value	Current execution priority.
dbgen	slave	Signal	Disallow (DAP) debugger access.
edbgrq	slave	Signal	External debug request.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE and corresponds to the RTL TXEV and RXEV signals.
fpudisable	slave	Signal	Configure core with no FPU on reset.
fpxxc	master	Value	Port that sends the value of the FPxxC exception flags (FPIXC, FPIDC, FPOFC, FPUFC, FPDZC, FPIOC). Each flag is present in the same position as in the FPSCR register (e.g. fpxxc[0] = FPIOC).
intisr	slave	Signal	This signal array delivers signals to the NVIC.
intnmi	slave	Signal	Configure non maskable interrupt.
lockup	master	Signal	Asserted when the processor is in lockup state.
mpudisable	slave	Signal	Configure core with no MPU on reset.
poreset	slave	Signal	Raising this signal will do a power-on reset of the core.
pv_ppbus_m	master	PVBus	The core will generate External Private Peripheral Bus requests on this port.
pvbus_m	master	PVBus	The core will generate bus requests on this port.
sleepdeep	master	Signal	Asserted when the processor is in deep sleep.
sleeping	master	Signal	Asserted when the processor is in sleep.
stcalib	slave	Value	This is the calibration value for the SysTick timer The following bit flag is modelled stcalib[25] - NOREF - Indicates no reference clock integrated. If stclk is not in use, set to 1
stclk	slave	ClockSignal	This is the reference clock for the SysTick timer.
sysreset	slave	Signal	Raising this signal will put the core into reset mode (but does not reset the debug logic).
sysresetreq	master	Signal	Asserted to indicate that a reset is required.
ticks	master	InstructionCount	Port allowing the number of instructions since startup to be read from the CPU.

## Parameters for ARMCortexM4CT

### BB\_PRESENT

Enable bitbanding.

Type: `bool`

Default value: `true`

### BIGENDINIT

Initialize processor to big endian mode.

Type: `bool`

Default value: `false`

### **DBGLVL**

0: No debug; 1: Minimal debug; 2: Full debug without DWT address matching; 3: Full debug support with, DWT can compare data as well as address.

Type: `uint8_t`

Default value: 3

### **LVL\_WIDTH**

Number of bits of interrupt priority.

Type: `uint8_t`

Default value: 3

### **NUM\_IRQ**

Number of user interrupts.

Type: `uint8_t`

Default value: 16

### **NUM\_MPU\_REGION**

Number of MPU regions.

Type: `uint8_t`

Default value: 8

### **TRACE\_LVL**

Level of instrumentation trace supported. 0: No trace, 1: Standard trace. ITM, TPIU and DWT triggers and counters present, 2: Full trace. ITM, TPIU, ETM and DWT triggers and counters present.

Type: `uint8_t`

Default value: 1

### **WIC**

Include support for WIC-mode deep sleep.

Type: `bool`

Default value: `true`



**cpi\_div**

divider for calculating CPI (Cycles Per Instruction).

Type: uint32\_t

Default value: 0x1

**cpi\_mul**

multiplier for calculating CPI (Cycles Per Instruction).

Type: uint32\_t

Default value: 0x1

**manager\_id**

Manager ID presented in bus transactions.

Type: uint64\_t

Default value: 0x0

**min\_sync\_level**

force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**reported\_patch\_level**

Purely cosmetic patch level value to be displayed from the CPUID register. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: int8\_t

Default value: -1

**reported\_revision\_number**

Purely cosmetic revision number value to be displayed from the CPUID register. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: int8\_t

Default value: -1

**semihosting-Thumb\_SVC**

T32 SVC number for semihosting.

Type: `uint8_t`

Default value: 171

### **semihosting-cmd\_line**

Command line available to semihosting SVC calls.

Type: `string`

Default value: N/A

### **semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **semihosting-enable**

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

Type: `bool`

Default value: `true`

### **semihosting-heap\_base**

Virtual address of heap base.

Type: `uint32_t`

Default value: 0x0

### **semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint32_t`

Default value: 0x20700000

### **semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint32_t`

Default value: 0x20800000

**semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint32_t`

Default value: `0x20700000`

**vfp-present**

Set whether the model has VFP support.

Type: `bool`

Default value: `true`

3.86 **ARMCortexM52CT**

Defined in `LISA/ARMCortexM52CT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Preliminary support

For an explanation of the quality levels, see [Quality level definitions](#).

**Changes in 11.31.15**

The following ports were removed:

- `auxfault`
- `currpri`

**About ARMCortexM52CT**

This model supports the M-Profile Vector Extension (MVE) and the Custom Datapath Extension (CDE). For more information, see [CDE](#).

**Iris and MTI instances for ARMCortexM52CT**

This model has the following Iris instances:

Name	Instance type
ARMCortexM52CT	ARM_Cortex-M52
ARMCortexM52CT.acp_mapper	PVBusMapper
ARMCortexM52CT.ext_bus	PVBusLogger
ARMCortexM52CT.ext_bus.mapper	PVBusMapper
ARMCortexM52CT.l1_incoherent_interconnect	PVCache
ARMCortexM52CT.l1_incoherent_interconnect.downstream[0].pvbusmaster	PVBusMaster

Name	Instance type
ARMCortexM52CT.11_incoherent_interconnect.upstream[Z] (where Z = 0-17)	PVBusSlave
ARMCortexM52CT.11dcache	PVCache
ARMCortexM52CT.11dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexM52CT.11dcache.upstream[0]	PVBusSlave
ARMCortexM52CT.11licache	PVCache
ARMCortexM52CT.11licache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexM52CT.11licache.upstream[0]	PVBusSlave
ARMCortexM52CT.12_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

Name	Component type
ARMCortexM52CT	ARM_Cortex-M52
ARMCortexM52CT.acp_mapper	PVBusMapper
ARMCortexM52CT.ext_bus	PVBusLogger
ARMCortexM52CT.ext_bus.mapper	PVBusMapper
ARMCortexM52CT.11_incoherent_interconnect	PVCache
ARMCortexM52CT.11_incoherent_interconnect.downstream[0].pvbusmaster	PVBusMaster
ARMCortexM52CT.11_incoherent_interconnect.upstream[Z] (where Z = 0-17)	PVBusSlave
ARMCortexM52CT.11dcache	PVCache
ARMCortexM52CT.11dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexM52CT.11dcache.upstream[0]	PVBusSlave
ARMCortexM52CT.11licache	PVCache
ARMCortexM52CT.11licache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexM52CT.11licache.upstream[0]	PVBusSlave
ARMCortexM52CT.12_flusher	AsyncCacheFlushUnit

### Ports for ARMCortexM52CT

Port	Direction	Protocol	Description
ahbd	slave	PVBus	Debug AHB - core bus slave driven by the DAP.
ahbp_m	master	PVBus	The core will generate Vendor System data accesses on this port.
ahbs	slave	PVBus	Slave AHB - External master (e.g. DMA) can write to TCMs (whether or not enabled in xTCMCR)
bigend	slave	Signal	Configure big endian data format.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions. Referred to in some documents as 'CLKIN'.
coproc_bus	slave	CoprocBusProtocol	Co-Processor Interface
core_dside_bus_gasket_in	slave	PVBus	Inbound D-side PVBus link is used when the external gasket is enabled. This allows externally processed D-side traffic to be fed back into the core path.
core_dside_bus_gasket_out	master	PVBus	Outbound D-side PVBus is exposed when has_core_dside_bus_gasket is true. This lets an external gasket observe or alter core data traffic before it re-enters on the inbound port.

Port	Direction	Protocol	Description
cpuwait	slave	Signal	Stall the CPU out of reset
currns	master	Signal	Current Security state of the processor
dbgen	slave	Signal	Invasive debug enable
dbgrestart	slave	Signal	Request for synchronised exit from halt mode
dbgrestorted	master	Signal	Handshakes with DBGRESTART
edbgrq	slave	Signal	External request to enter halt mode
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE and corresponds to the RTL TXEV and RXEV signals.
fpxxc	master	Value	Port that sends the value of the FPxxC exception flags (FPIX, FPID, FPOF, FPUF, FPDZ, FPIO). Each flag is present in the same position as in the FPSCR register (e.g. fpxxc[0] = FPIO).
halted	master	Signal	Indicates that the processor is in halt mode
idau_invalidate_region	slave	Value_64	64 bit number to invalid IDAU memory range (start_address<<32 end_address)
idau	master	PVBus	The core will generate IDAU requests on this port.
initnsvtor	slave	Value	Reset configuration port - Non-Secure Vector table offset (VTOR.TBLOFF[31:7]) out of reset This port remains functional no matter ARMv8-M Security Extensions are included or not When ARMv8-M Security Extensions are not included, all exceptions will use NS vector base address given by this port.
initpahben	slave	Signal	Enable P-AHB on the next reset
initsvtor	slave	Value	Reset configuration port - Secure Vector table offset (VTOR.TBLOFF[31:7]) out of reset It becomes functional when ARMv8-M Security Extensions are included When ARMv8-M Security Extensions are not included, this port will be ignored.
inittcmen	slave	Signal	Reset configuration port - TCM enable initialisation out of reset Bit[0] HIGH = ITCM is enabled Bit[1] HIGH = DTCM is enabled
intnum	master	Value	Exception number of the current execution context (from IPSR[8:0]) When the processor is in Thread mode, INTNUM is 0 When the processor is in Handler mode, INTNUM is the exception number of the currently-executing exception.
irq	slave	Signal	This signal array delivers signals to the NVIC.
lockdcaic	slave	Signal	Disable access to instruction cache direct cache access registers
lockdtgu	slave	Signal	Disable writes to registers associated with the DTGU
lockitgu	slave	Signal	Disable writes to registers associated with the ITGU
locknsmpu	slave	Signal	Disable writes to the Non-Secure MPU_*_NS registers
locknsvtor	slave	Signal	Cortex-M52-specific LOCKNSVTOR, LOCKSVTAIRCR, LOCKSMPU, LOCKNSMPU, LOCKSAU. Disable writes to VTOR_NS
lockpahb	slave	Signal	P-AHB related ports Disable writes to PAHBCR
locksau	slave	Signal	Disable writes to the SAU_* registers
locksmpu	slave	Signal	Disable writes to the Secure MPU_* registers
locksvtaircr	slave	Signal	Disable writes to VTOR_S, AIRCR.PRIS, AIRCR.BFHFNMINS
locktcm	slave	Signal	Disable writes to registers associated with the TCM
lockup	master	Signal	Asserted when the processor is in lockup state.
niden	slave	Signal	Non-invasive debug enable

Port	Direction	Protocol	Description
nmi	slave	Signal	Configure non maskable interrupt.
poreset	slave	Signal	Raising this signal will do a power-on reset of the core.
prequest	slave	PChannel	Low Power Interface
pv_ppbus_m	master	PVBus	The core will generate External Private Peripheral Bus requests on this port.
pvbus_m	master	PVBus	The core will generate bus requests on this port.
qrequest	slave	PChannel	Low Power Transition handshake interface
sleepdeep	master	Signal	Asserted when the processor is in deep sleep.
sleeping	master	Signal	Asserted when the processor is in sleep.
spiden	slave	Signal	Secure invasive debug enable
spniden	slave	Signal	Secure non-invasive debug enable
stcalib	slave	Value	These are the secure and non-secure calibration values for the SysTick timer, called CFGSSTCALIB/CFGNSSTCALIB in the M52 specification. stcalib[0] is the secure timer calibration, stcalib[1] is the non-secure timer calibration. In the case where the Security Extension is not present, stcalib[0] should be used. The following bit flag is modelled for each stcalib[*][25] - NOREF - Indicates no reference clock integrated. If stclk is not in use, set to 1
stclk	slave	ClockSignal	This is the reference clock for the SysTick timer, called SSTCLKEN/NSSTCLKEN in the M52 specification. This clock is provided for both secure and non secure inputs, even where ARMv8-M Security Extensions are included.
sysreset	slave	Signal	Raising this signal will put the core into reset mode (but does not reset the debug logic).
sysresetreq	master	Signal	Asserted to indicate that a reset is required.
ticks	master	InstructionCount	Port allowing the number of instructions since startup to be read from the CPU.
wicenack	master	Signal	Acknowledge signal for WICENREQ
wicenreq	slave	Signal	Request for deep sleep to be WIC-based deep sleep.
wicsense	master	Signal	Indicates which input events can cause the WIC to generate the WAKEUP signal.

## Parameters for ARMCortexM52CT

### CDEMAPPEDONCP

Bit N specifies whether the instruction for coprocessor N (CP7:CP0) is redirected to the CDE module.

Type: uint8\_t

Default value: 255

### CDERTLID

Value of ID\_AFR0.CDERTLID.

Type: uint8\_t

Default value: 32

**CFGBIGEND**

Initialize processor to big endian mode.

Type: `bool`

Default value: `false`

**CFGCPUINST**

CPU instance number. This is part of the TCM base address, in bits 25:24.

Type: `uint8_t`

Default value: 0

**CFGDTCMSZ**

Size of the data TCM. 0=No DTCM implemented. Otherwise=Size of DTCM= $\text{pow}(2, \text{CFGDTCMSZ} - 1)$  KB. Minimum size is 4KB.

Type: `uint8_t`

Default value: 9

**CFGITCMSZ**

Size of the instruction TCM. 0=No ITCM implemented. Otherwise=Size of ITCM= $\text{pow}(2, \text{CFGITCMSZ} - 1)$  KB. Minimum size is 4KB.

Type: `uint8_t`

Default value: 9

**CFGMEMALIAS**

Memory address alias bit for the ITCM, DTCM and P-AHB regions. 0=No alias, 1=Alias bit 26, 2=Alias bit 27, 4=Alias bit 28.

Type: `uint8_t`

Default value: 0

**CFGNOCDECP**

Bit N means external coprocessor N (CP7:CP0) disable for CDE coprocessor.

Type: `uint8_t`

Default value: 0

**CFGPAACBTI**

Enables support for the Pointer Authentication and Branch Target Identification (PACBTI) Extension. FALSE: Disabled, TRUE: PAC implemented using the QARMA3 algorithm with BTI.

Type: `bool`

Default value: `false`

**CFGPAHBSZ**

Size of the P-AHB peripheral port memory region. 0=P-AHB disabled, 1=64MB, 2=128MB, 3=256MB, 4=512MB.

Type: `uint8_t`

Default value: 0

**CPIF**

Specifies whether the external coprocessor interface is included.

Type: `bool`

Default value: `true`

**CPNSPRESENT**

Bit N means external coprocessor N (CP7:CP0) is accessible in Non-Secure state.

Type: `uint8_t`

Default value: 255

**CPSPPRESENT**

Bit N means external coprocessor N (CP7:CP0) is accessible in Secure state.

Type: `uint8_t`

Default value: 255

**CTI**

CTI (Cross Trigger Interface) included.

Type: `bool`

Default value: `false`

**CTI\_irq0\_pin**

CTI interrupt request 0 pin.



Type: `uint16_t`

Default value: 4

### **CTI\_irq1\_pin**

CTI interrupt request 1 pin.

Type: `uint16_t`

Default value: 5

### **DBG\_LVL**

0: Minimal debug; 1: 2 Watchpoints, 4 Breakpoint comparators; 2: 4 Watchpoints, 8 Breakpoint comparators; 3: 8 Watchpoints, 8 Breakpoint comparators.

Type: `uint8_t`

Default value: 2

### **DCACHESZ**

Whether the D-cache is included and, if included, the size of it. Bit 0: 0=No D-cache included, 1=D-cache included. Bits [4:1]: 0x0=4KB D-cache, 0x1=8KB D-cache, 0x3=16KB D-cache, 0x7=32KB D-cache, 0xF=64KB D-cache.

Type: `uint8_t`

Default value: 15

### **DTGU**

DTCM Security Gate Unit included.

Type: `bool`

Default value: `false`

### **DTGUBLKSZ**

DTCM gate unit block size. Size= $\text{pow}(2, \text{DTGUBLKSZ} + 5)$  bytes.

Type: `uint8_t`

Default value: 3

### **DTGUMAXBLKS**

Maximum number of DTCM gate unit blocks. Number of blocks= $\text{pow}(2, \text{DTGUMAXBLKS})$ .

Type: `uint8_t`

Default value: 0

**ECOREVNUM**

ECO Revision number.

Type: `uint64_t`

Default value: `0x0`

**ERRDEVID.NUM**

RAS: Number of implemented error record indexes, 0 to 1.

Type: `uint8_t`

Default value: `1`

**ETM**

Support for ETM trace. `false` : No ETM trace included, `true`: ETM trace included.

Type: `bool`

Default value: `true`

**FPMVE**

Set whether the model has FP and / or MVE support. 0: No FP and MVE support. 1: FP half and single precision. 2: FP half, single and double precision. 3: MVE integer. 4: FP half and single precision and MVE integer. 5: FP half, single and double precision and MVE floating point.

Type: `uint8_t`

Default value: `5`

**ICACHESZ**

Whether the I-cache is included and, if included, the size of it. Bit 0: 0=No I-cache included, 1=I-cache included. Bits [6:1]: `0x0`=1KB I-cache (only with unified cache), `0x1`=2KB I-cache (only with unified cache), `0x3`=4KB I-cache, `0x7`=8KB I-cache, `0xF`=16KB I-cache, `0x1F`=32KB I-cache, `0x3F`=64KB I-cache.

Type: `uint8_t`

Default value: `63`

**INITNSVTOR**

Non-Secure vector-table offset at reset.

Type: `uint32_t`

Default value: `0x0`

**INITPAHBEN**

The P-AHB enable state at reset.

Type: `bool`

Default value: `false`

**INITSVTOR**

Secure vector-table offset at reset.

Type: `uint32_t`

Default value: `0x0`

**INITTCMEN**

The TCM enable state at reset. Bit 0 corresponds to ITCM enable state, bit 1 corresponds to DTCM enable state.

Type: `uint8_t`

Default value: `3`

**IRQDIS0**

IRQ line disable mask. Bit *n* of this 32-bit parameter disables IRQ[*n*+0].

Type: `uint32_t`

Default value: `0x0`

**IRQDIS1**

IRQ line disable mask. Bit *n* of this 32-bit parameter disables IRQ[*n*+32].

Type: `uint32_t`

Default value: `0x0`

**IRQDIS10**

IRQ line disable mask. Bit *n* of this 32-bit parameter disables IRQ[*n*+320].

Type: `uint32_t`

Default value: `0x0`

**IRQDIS11**

IRQ line disable mask. Bit *n* of this 32-bit parameter disables IRQ[*n*+352].

Type: `uint32_t`

Default value: 0x0

**IRQDIS12**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+384].

Type: uint32\_t

Default value: 0x0

**IRQDIS13**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+416].

Type: uint32\_t

Default value: 0x0

**IRQDIS14**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+448].

Type: uint32\_t

Default value: 0x0

**IRQDIS2**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+64].

Type: uint32\_t

Default value: 0x0

**IRQDIS3**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+96].

Type: uint32\_t

Default value: 0x0

**IRQDIS4**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+128].

Type: uint32\_t

Default value: 0x0

**IRQDIS5**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+160].

Type: uint32\_t

Default value: 0x0

**IRQDIS6**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+192].

Type: uint32\_t

Default value: 0x0

**IRQDIS7**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+224].

Type: uint32\_t

Default value: 0x0

**IRQDIS8**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+256].

Type: uint32\_t

Default value: 0x0

**IRQDIS9**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+288].

Type: uint32\_t

Default value: 0x0

**IRQLVL**

Number of bits of interrupt priority.

Type: uint8\_t

Default value: 3

**ITGU**

ITCM Security Gate Unit included.

Type: bool

Default value: false

**ITGUBLKSZ**

ITCM gate unit block size. Size=pow(2, ITGUBLKSZ + 5) bytes.

Type: uint8\_t

Default value: 3

**ITGUMAXBLKS**

Maximum number of ITCM gate unit blocks. Number of blocks= $\text{pow}(2, \text{ITGUMAXBLKS})$ .

Type: `uint8_t`

Default value: 0

**ITM**

Level of instrumentation trace supported. `false` : No ITM trace included, `true`: ITM trace included.

Type: `bool`

Default value: `true`

**IWIC**

Include support for Internal Wake-up Interrupt Controller.

Type: `bool`

Default value: `true`

**LOCKDTGU**

Lock down of Data TGU registers write.

Type: `bool`

Default value: `false`

**LOCKITGU**

Lock down of Instruction TGU registers write.

Type: `bool`

Default value: `false`

**LOCKTCM**

Lock down of TCM registers write.

Type: `bool`

Default value: `false`

**LOCK\_NS\_MPU**

Lock down of Non-Secure MPU registers write.

Type: `bool`

Default value: `false`

### **LOCK\_SAU**

Lock down of SAU registers write.

Type: `bool`

Default value: `false`

### **LOCK\_S\_MPU**

Lock down of Secure MPU registers write.

Type: `bool`

Default value: `false`

### **MPU\_NS**

Number of regions in the Non-Secure MPU. If Security Extensions are absent, this is the total number of MPU regions.

Type: `uint8_t`

Default value: 8

### **MPU\_S**

Number of regions in the Secure MPU. If Security Extensions are absent, this is ignored.

Type: `uint8_t`

Default value: 8

### **NUMIRQ**

Number of user interrupts.

Type: `uint16_t`

Default value: 32

### **SAU**

Number of SAU regions (0 => no SAU).

Type: `uint8_t`

Default value: 4

### **SECEXT**

Whether the ARMv8-M Security Extensions are included.

Type: `bool`

Default value: `true`

### **UCACHE**

Whether the I-cache acts as a unified cache (ICACHESZ is used for the size).

Type: `bool`

Default value: `false`

### **WICLINES**

Number of lines supported by the WIC interface.

Type: `uint16_t`

Default value: `35`

### **cde\_impl\_name**

Name of the CDE implementation for this core (implementation contributed by MTI plugin).

Type: `string`

Default value: `N/A`

### **cpi\_div**

divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

### **cpi\_mul**

multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

### **dcache-state\_modelled**

Set whether D-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **delay\_faultmask\_update**

Delay FAULTMASK update to context sync.



Type: `bool`

Default value: `false`

### **delay\_sysreg\_update**

Delay some system register updates (e.g. SHCSR) to context sync.

Type: `bool`

Default value: `false`

### **ecc\_on**

Enable Error Correcting Code.

Type: `bool`

Default value: `false`

### **has\_cde**

Enables Custom Datapath Extensions.

Type: `bool`

Default value: `false`

### **has\_core\_dside\_bus\_gasket**

STL gasket enabled.

Type: `bool`

Default value: `false`

### **icache-state\_modelled**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **manager\_id**

Manager ID presented in bus transactions.

Type: `uint64_t`

Default value: `0x0`

### **min\_sync\_level**

force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: 0

### **reported\_patch\_level**

Purely cosmetic patch level value to be displayed from the CPUID register. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: `int8_t`

Default value: -1

### **reported\_revision\_number**

Purely cosmetic revision number value to be displayed from the CPUID register. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: `int8_t`

Default value: -1

### **semihosting-Thumb\_SVC**

T32 SVC number for semihosting.

Type: `uint8_t`

Default value: 171

### **semihosting-cmd\_line**

Command line available to semihosting SVC calls.

Type: `string`

Default value: N/A

### **semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **semihosting-enable**

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

Type: `bool`

Default value: `true`

**semihosting-heap\_base**

Virtual address of heap base.

Type: `uint32_t`

Default value: `0x0`

**semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint32_t`

Default value: `0x20700000`

**semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint32_t`

Default value: `0x20800000`

**semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint32_t`

Default value: `0x20700000`

**trace\_style**

MVE instruction trace style: Add 16 for `[*-]` beat trace. Add 32 for tracing IMPLIED LOB instructions. Add 64 to change opcode of implied BF to `0xBF00`.

Type: `uint8_t`

Default value: `2`

**vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

## 3.87 ARMCortexM55CT

Defined in `LISA/ARMCortexM55CT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r1p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following ports were added:

- `invalidate_rse_ecc`

The following ports were removed:

- `auxfault`
- `currpri`
- `tramdueeca`
- `tramsceeca`

### About ARMCortexM55CT

The model supports Custom Datapath Extension (CDE). For more information, see [CDE](#).

The model does not support the following functionality:

- Programmable MBIST controller (PMC-100).
- Error Correcting Code (ECC).
- Q-Channel.

The following interfaces and registers are not modeled:

- ITM and ETM trace and trace synchronization and trigger interface signals.
- Dual-core lock-step operation.
- Interrupt latencies.
- Prefetcher Control Register (PFCR).
- Direct cache access registers.

In the Memory System Control Register (MSCR), only the cache-related bits are supported:

- `FORCEWT`
- `DCACTIVE`
- `ICACTIVE`

- IDCCLEAN

## Differences between the model and the RTL

In hardware, `PMU_CCNTR` is an alias of the `DWT_CYCCNT` register, so they contain the same values. In the model, `PMU_CCNTR` is implemented differently to `DWT_CYCCNT`, so they contain different values. The value held in `DWT_CYCCNT` is not representative of hardware.

## Implementation of ITM in M-class models

This model has a parameter that enables partial support for Instrumentation Trace Macrocell (ITM). In hardware or RTL, trace data from ITM is sent in packets to the trace block serially using a single pin or wire. In the model, if it is enabled, the ITM trace data is output using an MTI trace source called ITM. The ITM trace source has an `ITM_PACKET_TYPE` field. The following table shows which packet types the model supports:

Field value	Description	Supported by model
ITM_SYNC	Synchronization packet	Not supported.
ITM_P_OVERFLOW	Protocol: Overflow packet	Not supported.
ITM_P_LOCAL_TIMESTAMP	Protocol: Local timestamp packets	Not supported.
ITM_P_GLOBAL_TIMESTAMP	Protocol: Global timestamp packets	Not supported.
ITM_P_EXTEN	Protocol: Extension packet	Not supported.
ITM_S_INSTRUMENTATION	Source: Instrumentation packet	Supported.
ITM_S_DWT_EVENT_COUNTER	Hardware source: Event counter wrapping	Not supported.
ITM_S_DWT_EXCEPTION	Hardware source: Exception tracing	Supported.
ITM_S_DWT_PC_SAMPLING	Hardware source: PC sampling	Not supported.
ITM_S_DWT_DATA_PC_TRACE	Hardware source: DWT Data trace PC value	Supported.
ITM_S_DWT_DATA_ADDRESS_TRACE	Hardware source: DWT Data trace address value	Supported.
ITM_S_DWT_DATA_DATA_TRACE	Hardware source: DWT Data trace DATA value	Supported.

## Iris and MTI instances for ARMCortexM55CT

This model has the following Iris instances:

Name	Instance type
ARMCortexM55CT	ARM_Cortex-M55
ARMCortexM55CT.acp_mapper	PVBusMapper
ARMCortexM55CT.ext_bus	PVBusLogger
ARMCortexM55CT.ext_bus.mapper	PVBusMapper
ARMCortexM55CT.l1_incoherent_interconnect	PVCache
ARMCortexM55CT.l1_incoherent_interconnect.downstream[0].pvbusmaster	PVBusMaster
ARMCortexM55CT.l1_incoherent_interconnect.upstream[Z] (where Z = 0-17)	PVBusSlave
ARMCortexM55CT.l1dcache	PVCache
ARMCortexM55CT.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexM55CT.l1dcache.upstream[0]	PVBusSlave
ARMCortexM55CT.l1icache	PVCache

Name	Instance type
ARMCortexM55CT.l1licache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexM55CT.l1licache.upstream[0]	PVBusSlave
ARMCortexM55CT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

Name	Component type
ARMCortexM55CT	ARM_Cortex-M55
ARMCortexM55CT.acp_mapper	PVBusMapper
ARMCortexM55CT.ext_bus	PVBusLogger
ARMCortexM55CT.ext_bus.mapper	PVBusMapper
ARMCortexM55CT.l1_incoherent_interconnect	PVCache
ARMCortexM55CT.l1_incoherent_interconnect.downstream[0].pvbusmaster	PVBusMaster
ARMCortexM55CT.l1_incoherent_interconnect.upstream[Z] (where Z = 0–17)	PVBusSlave
ARMCortexM55CT.l1dcache	PVCache
ARMCortexM55CT.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexM55CT.l1dcache.upstream[0]	PVBusSlave
ARMCortexM55CT.l1licache	PVCache
ARMCortexM55CT.l1licache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexM55CT.l1licache.upstream[0]	PVBusSlave
ARMCortexM55CT.l2_flusher	AsyncCacheFlushUnit

## Ports for ARMCortexM55CT

Port	Direction	Protocol	Description
ahbd	slave	PVBus	Debug AHB - core bus slave driven by the DAP.
ahbp_m	master	PVBus	The core will generate Vendor System data accesses on this port.
ahbs	slave	PVBus	Slave AHB - External master (e.g. DMA) can write to TCMs (whether or not enabled in xTCMCR)
bigend	slave	Signal	Configure big endian data format.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions. Referred to in some documents as 'CLKIN'.
coproc_bus	slave	CoprocBusProtocol	Co-Processor Interface
core_dside_bus_gasket_in	slave	PVBus	Inbound D-side PVBus link is used when the external gasket is enabled. This allows externally processed D-side traffic to be fed back into the core path.
core_dside_bus_gasket_out	master	PVBus	Outbound D-side PVBus is exposed when has_core_dside_bus_gasket is true. This lets an external gasket observe or alter core data traffic before it re-enters on the inbound port.
cpuwait	slave	Signal	Stall the CPU out of reset
currns	master	Signal	Current Security state of the processor
dbgen	slave	Signal	Invasive debug enable
dbgrestart	slave	Signal	Request for synchronised exit from halt mode
dbgrestarted	master	Signal	Handshakes with DBGRESTART

Port	Direction	Protocol	Description
edbgrq	slave	Signal	External request to enter halt mode
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE and corresponds to the RTL TXEV and RXEV signals.
fpxxc	master	Value	Port that sends the value of the FPxxC exception flags (FPIXIC, FPIDC, FPOFC, FPUFC, FPDZC, FPIOC). Each flag is present in the same position as in the FPSCR register (e.g. fpxxc[0] = FPIOC).
halted	master	Signal	Indicates that the processor is in halt mode
idau_invalidate_region	slave	Value_64	64 bit number to invalid IDAU memory range (start_address<<32 end_address)
idau	master	PVBus	The core will generate IDAU requests on this port.
initnsvtor	slave	Value	Reset configuration port - Non-Secure Vector table offset (VTOR.TBLOFF[31:7]) out of reset This port remains functional no matter ARMv8-M Security Extensions are included or not When ARMv8-M Security Extensions are not included, all exceptions will use NS vector base address given by this port.
initpahben	slave	Signal	Enable P-AHB on the next reset
initsvtor	slave	Value	Reset configuration port - Secure Vector table offset (VTOR.TBLOFF[31:7]) out of reset It becomes functional when ARMv8-M Security Extensions are included When ARMv8-M Security Extensions are not included, this port will be ignored.
inittcm	slave	Signal	Reset configuration port - TCM enable initialisation out of reset Bit[0] HIGH = ITCM is enabled Bit[1] HIGH = DTCM is enabled
intnum	master	Value	Exception number of the current execution context (from IPSR[8:0]) When the processor is in Thread mode, INTNUM is 0 When the processor is in Handler mode, INTNUM is the exception number of the currently-executing exception.
invalidate_rse_ecc	slave	Signal	Invalidate DTCM ECC tags because TrustedRAM keys changed
irq	slave	Signal	This signal array delivers signals to the NVIC.
lockdcaic	slave	Signal	Disable access to instruction cache direct cache access registers
lockdtgu	slave	Signal	Disable writes to registers associated with the DTGU
lockitgu	slave	Signal	Disable writes to registers associated with the ITGU
locknsmpu	slave	Signal	Disable writes to the Non-Secure MPU_*_NS registers
locknsvtor	slave	Signal	Cortex-M55-specific LOCKNSVTOR, LOCKSVTAIRCR, LOCKSMPU, LOCKNSMPU, LOCKSAU. Disable writes to VTOR_NS
lockpahb	slave	Signal	P-AHB related ports Disable writes to PAHBSCR
locksau	slave	Signal	Disable writes to the SAU_* registers
locksmpu	slave	Signal	Disable writes to the Secure MPU_* registers
locksvtaircr	slave	Signal	Disable writes to VTOR_S, AIRCR.PRIS, AIRCR.BFHFNMINS
locktcm	slave	Signal	Disable writes to registers associated with the TCM
lockup	master	Signal	Asserted when the processor is in lockup state.
niden	slave	Signal	Non-invasive debug enable
nmi	slave	Signal	Configure non maskable interrupt.
poreset	slave	Signal	Raising this signal will do a power-on reset of the core.
prequest	slave	PChannel	Low Power Interface
pv_ppbus_m	master	PVBus	The core will generate External Private Peripheral Bus requests on this port.

Port	Direction	Protocol	Description
pvbus_m	master	PVBus	The core will generate bus requests on this port.
qrequest	slave	PChannel	Low Power Transition handshake interface
sleepdeep	master	Signal	Asserted when the processor is in deep sleep.
sleeping	master	Signal	Asserted when the processor is in sleep.
spiden	slave	Signal	Secure invasive debug enable
spniden	slave	Signal	Secure non-invasive debug enable
stcalib	slave	Value	These are the secure and non-secure calibration values for the SysTick timer, called CFGSSTCALIB/CFGNSSTCALIB in the M55 specification. stcalib[0] is the secure timer calibration, stcalib[1] is the non-secure timer calibration. In the case where the Security Extension is not present, stcalib[0] should be used. The following bit flag is modelled for each stcalib[*][25] - NOREF - Indicates no reference clock integrated. If stclk is not in use, set to 1
stclk	slave	ClockSignal	This is the reference clock for the SysTick timer, called SSTCLKEN/NSSTCLKEN in the M55 specification. This clock is provided for both secure and non secure inputs, even where ARMv8-M Security Extensions are included.
sysreset	slave	Signal	Raising this signal will put the core into reset mode (but does not reset the debug logic).
sysresetreq	master	Signal	Asserted to indicate that a reset is required.
ticks	master	InstructionCount	Port allowing the number of instructions since startup to be read from the CPU.
wicenack	master	Signal	Acknowledge signal for WICENREQ
wicenreq	slave	Signal	Request for deep sleep to be WIC-based deep sleep.
wicsense	master	Signal	Indicates which input events can cause the WIC to generate the WAKEUP signal.

## Parameters for ARM CortexM55CT

### CDEMAPPEDONCP

Bit N specifies whether the instruction for coprocessor N (CP7:CP0) is redirected to the CDE module.

Type: uint8\_t

Default value: 255

### CDERTLID

Value of ID\_AFR0.CDERTLID.

Type: uint8\_t

Default value: 32

### CFGBIGEND

Initialize processor to big endian mode.



Type: `bool`

Default value: `false`

### **CFGDTCMSZ**

Size of the data TCM. 0=No DTCM implemented. Otherwise=Size of DTCM= $\text{pow}(2, \text{CFGDTCMSZ} - 1)$  KB. Minimum size is 4KB.

Type: `uint8_t`

Default value: 9

### **CFGITCMSZ**

Size of the instruction TCM. 0=No ITCM implemented. Otherwise=Size of ITCM= $\text{pow}(2, \text{CFGITCMSZ} - 1)$  KB. Minimum size is 4KB.

Type: `uint8_t`

Default value: 9

### **CFGMEMALIAS**

Memory address alias bit for the ITCM, DTCM and P-AHB regions. 0=No alias, 1=Alias bit 24, 2=Alias bit 25, 4=Alias bit 26, 8=Alias bit 27, 16=Alias bit 28.

Type: `uint8_t`

Default value: 0

### **CFGNOCDECP**

Bit N means external coprocessor N (CP7:CP0) disable for CDE coprocessor.

Type: `uint8_t`

Default value: 0

### **CFGPAHBSZ**

Size of the P-AHB peripheral port memory region. 0=P-AHB disabled, 1=64MB, 2=128MB, 3=256MB, 4=512MB.

Type: `uint8_t`

Default value: 0

### **CPIF**

Specifies whether the external coprocessor interface is included.

Type: `bool`

Default value: `true`

**CPNSPRESENT**

Bit N means external coprocessor N (CP7:CP0) is accessible in Non-Secure state.

Type: `uint8_t`

Default value: 255

**CPSPPRESENT**

Bit N means external coprocessor N (CP7:CP0) is accessible in Secure state.

Type: `uint8_t`

Default value: 255

**CTI**

CTI (Cross Trigger Interface) included.

Type: `bool`

Default value: `false`

**CTI\_irq0\_pin**

CTI interrupt request 0 pin.

Type: `uint16_t`

Default value: 4

**CTI\_irq1\_pin**

CTI interrupt request 1 pin.

Type: `uint16_t`

Default value: 5

**DBGLVL**

0: Minimal debug; 1: 2 Watchpoints, 4 Breakpoint comparators; 2: 4 Watchpoints, 8 Breakpoint comparators; 3: 8 Watchpoints, 8 Breakpoint comparators.

Type: `uint8_t`

Default value: 2

**DCACHESZ**

Whether the D-cache is included and, if included, the size of it. Bit 0: 0=No D-cache included, 1=D-cache included. Bits [4:1]: 0x0=4KB D-cache, 0x1=8KB D-cache, 0x3=16KB D-cache, 0x7=32KB D-cache, 0xF=64KB D-cache.

Type: `uint8_t`

Default value: 15

**DTGU**

DTCM Security Gate Unit included.

Type: `bool`

Default value: `false`

**DTGUBLKSZ**

DTCM gate unit block size. Size= $\text{pow}(2, \text{DTGUBLKSZ} + 5)$  bytes.

Type: `uint8_t`

Default value: 3

**DTGUMAXBLKS**

Maximum number of DTCM gate unit blocks. Number of blocks= $\text{pow}(2, \text{DTGUMAXBLKS})$ .

Type: `uint8_t`

Default value: 0

**ECOREVNUM**

ECO Revision number.

Type: `uint64_t`

Default value: 0x0

**ERRDEVID.NUM**

RAS: Number of implemented error record indexes, 0 to 1.

Type: `uint8_t`

Default value: 1

**ETM**

Support for ETM trace. `false` : No ETM trace included, `true`: ETM trace included.

Type: `bool`

Default value: `true`

### **FPU**

Set whether the model has VFP support.

Type: `bool`

Default value: `true`

### **ICACHESZ**

Whether the I-cache is included and, if included, the size of it. Bit 0: 0=No I-cache included, 1=I-cache included. Bits [4:1]: 0x0=4KB I-cache, 0x1=8KB I-cache, 0x3=16KB I-cache, 0x7=32KB I-cache, 0xF=64KB I-cache.

Type: `uint8_t`

Default value: 15

### **INITNSVTOR**

Non-Secure vector-table offset at reset.

Type: `uint32_t`

Default value: 0x0

### **INITPAHBEN**

The P-AHB enable state at reset.

Type: `bool`

Default value: `false`

### **INITSVTOR**

Secure vector-table offset at reset.

Type: `uint32_t`

Default value: 0x0

### **INITTCMEN**

The TCM enable state at reset. Bit 0 corresponds to ITCM enable state, bit 1 corresponds to DTCM enable state.

Type: `uint8_t`

Default value: 3

**IRQDIS0**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+0].

Type: `uint32_t`

Default value: 0x0

**IRQDIS1**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+32].

Type: `uint32_t`

Default value: 0x0

**IRQDIS10**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+320].

Type: `uint32_t`

Default value: 0x0

**IRQDIS11**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+352].

Type: `uint32_t`

Default value: 0x0

**IRQDIS12**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+384].

Type: `uint32_t`

Default value: 0x0

**IRQDIS13**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+416].

Type: `uint32_t`

Default value: 0x0

**IRQDIS14**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+448].

Type: `uint32_t`

Default value: 0x0

**IRQDIS2**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+64].

Type: `uint32_t`

Default value: `0x0`

**IRQDIS3**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+96].

Type: `uint32_t`

Default value: `0x0`

**IRQDIS4**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+128].

Type: `uint32_t`

Default value: `0x0`

**IRQDIS5**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+160].

Type: `uint32_t`

Default value: `0x0`

**IRQDIS6**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+192].

Type: `uint32_t`

Default value: `0x0`

**IRQDIS7**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+224].

Type: `uint32_t`

Default value: `0x0`

**IRQDIS8**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+256].

Type: `uint32_t`

Default value: `0x0`

**IRQDIS9**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+288].

Type: `uint32_t`

Default value: `0x0`

**IRQLVL**

Number of bits of interrupt priority.

Type: `uint8_t`

Default value: `3`

**ITGU**

ITCM Security Gate Unit included.

Type: `bool`

Default value: `false`

**ITGUBLKSZ**

ITCM gate unit block size. Size= $\text{pow}(2, \text{ITGUBLKSZ} + 5)$  bytes.

Type: `uint8_t`

Default value: `3`

**ITGUMAXBLKS**

Maximum number of ITCM gate unit blocks. Number of blocks= $\text{pow}(2, \text{ITGUMAXBLKS})$ .

Type: `uint8_t`

Default value: `0`

**ITM**

Level of instrumentation trace supported. `false` : No ITM trace included, `true`: ITM trace included.

Type: `bool`

Default value: `true`

**IWIC**

Include support for Internal Wake-up Interrupt Controller.

Type: `bool`

Default value: `true`

**LOCKDTGU**

Lock down of Data TGU registers write.

Type: `bool`

Default value: `false`

**LOCKITGU**

Lock down of Instruction TGU registers write.

Type: `bool`

Default value: `false`

**LOCKTCM**

Lock down of TCM registers write.

Type: `bool`

Default value: `false`

**LOCK\_NS\_MPU**

Lock down of Non-Secure MPU registers write.

Type: `bool`

Default value: `false`

**LOCK\_SAU**

Lock down of SAU registers write.

Type: `bool`

Default value: `false`

**LOCK\_S\_MPU**

Lock down of Secure MPU registers write.

Type: `bool`

Default value: `false`

**MPU\_NS**

Number of regions in the Non-Secure MPU. If Security Extensions are absent, this is the total number of MPU regions.

Type: `uint8_t`



Default value: 8

**MPU\_S**

Number of regions in the Secure MPU. If Security Extensions are absent, this is ignored.

Type: `uint8_t`

Default value: 8

**MVE**

Set whether the model has MVE support. If FPU = 0: 0=MVE not included, 1=Integer subset of MVE included. If FPU = 1: 0=MVE not included, 1=Integer subset of MVE included, 2=Integer and half and single precision floating point MVE included.

Type: `uint8_t`

Default value: 1

**NUMIRQ**

Number of user interrupts.

Type: `uint16_t`

Default value: 32

**SAU**

Number of SAU regions (0 => no SAU).

Type: `uint8_t`

Default value: 4

**SECEXT**

Whether the ARMv8-M Security Extensions are included.

Type: `bool`

Default value: `true`

**WICLINES**

Number of lines supported by the WIC interface.

Type: `uint16_t`

Default value: 35

**cde\_impl\_name**

Name of the CDE implementation for this core (implementation contributed by MTI plugin).

Type: `string`

Default value: `N/A`

**cpi\_div**

divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**cpi\_mul**

multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**dcache-state\_modelled**

Set whether D-cache has stateful implementation.

Type: `bool`

Default value: `false`

**delay\_faultmask\_update**

Delay FAULTMASK update to context sync.

Type: `bool`

Default value: `false`

**delay\_sysreg\_update**

Delay some system register updates (e.g. SHCSR) to context sync.

Type: `bool`

Default value: `false`

**ecc\_on**

Enable Error Correcting Code.

Type: `bool`

Default value: `false`

**has\_cde**

Enables Custom Datapath Extensions.

Type: `bool`

Default value: `false`

**has\_core\_dside\_bus\_gasket**

STL gasket enabled.

Type: `bool`

Default value: `false`

**icache-state\_modelled**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

**manager\_id**

Manager ID presented in bus transactions.

Type: `uint64_t`

Default value: `0x0`

**min\_sync\_level**

force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

**reported\_patch\_level**

Purely cosmetic patch level value to be displayed from the CPUID register. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: `int8_t`

Default value: `-1`

**reported\_revision\_number**

Purely cosmetic revision number value to be displayed from the CPUID register. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: `int8_t`

Default value: -1

### **`rse_ecc_support`**

Support for ECC initialization in TCM, following RSE spec. 0=No support, 1=Supported, faults are reported as uncorrected errors.

Type: `uint8_t`

Default value: 0

### **`semihosting-Thumb_SVC`**

T32 SVC number for semihosting.

Type: `uint8_t`

Default value: 171

### **`semihosting-cmd_line`**

Command line available to semihosting SVC calls.

Type: `string`

Default value: N/A

### **`semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **`semihosting-enable`**

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

Type: `bool`

Default value: `true`

### **`semihosting-heap_base`**

Virtual address of heap base.

Type: `uint32_t`

Default value: 0x0

**semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint32_t`

Default value: `0x20700000`

**semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint32_t`

Default value: `0x20800000`

**semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint32_t`

Default value: `0x20700000`

**trace\_style**

MVE instruction trace style: Add 16 for `[*-]` beat trace. Add 32 for tracing IMPLIED LOB instructions. Add 64 to change opcode of implied BF to `0xBF00`.

Type: `uint8_t`

Default value: `2`

**vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

3.88 ARMCortexM7CT

Defined in `LISA/ARMCortexM7CT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r1p2	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

## Changes in 11.31.15

The following ports were removed:

- `auxfault`
- `currpri`

## Differences between the model and the RTL

- TCMs are modeled internally and the model does not support external TCMs or the ports associated with them.
- ECC support is hardware-specific so is not modeled.

## Implementation of ITM in M-class models

This model has a parameter that enables partial support for Instrumentation Trace Macrocell (ITM). In hardware or RTL, trace data from ITM is sent in packets to the trace block serially using a single pin or wire. In the model, if it is enabled, the ITM trace data is output using an MTI trace source called ITM. The ITM trace source has an `ITM_PACKET_TYPE` field. The following table shows which packet types the model supports:

Field value	Description	Supported by model
ITM_SYNC	Synchronization packet	Not supported.
ITM_P_OVERFLOW	Protocol: Overflow packet	Not supported.
ITM_P_LOCAL_TIMESTAMP	Protocol: Local timestamp packets	Not supported.
ITM_P_GLOBAL_TIMESTAMP	Protocol: Global timestamp packets	Not supported.
ITM_P_EXTEN	Protocol: Extension packet	Not supported.
ITM_S_INSTRUMENTATION	Source: Instrumentation packet	Supported.
ITM_S_DWT_EVENT_COUNTER	Hardware source: Event counter wrapping	Not supported.
ITM_S_DWT_EXCEPTION	Hardware source: Exception tracing	Supported.
ITM_S_DWT_PC_SAMPLING	Hardware source: PC sampling	Not supported.
ITM_S_DWT_DATA_PC_TRACE	Hardware source: DWT Data trace PC value	Supported.
ITM_S_DWT_DATA_ADDRESS_TRACE	Hardware source: DWT Data trace address value	Supported.
ITM_S_DWT_DATA_DATA_TRACE	Hardware source: DWT Data trace DATA value	Supported.

## Iris and MTI instances for ARM CortexM7CT

This model has the following Iris instances:

Name	Instance type
ARM CortexM7CT	ARM_Cortex-M7
ARM CortexM7CT.acp_mapper	PVBusMapper
ARM CortexM7CT.ext_bus	PVBusLogger
ARM CortexM7CT.ext_bus.mapper	PVBusMapper
ARM CortexM7CT.l1_incoherent_interconnect	PVCache
ARM CortexM7CT.l1_incoherent_interconnect.downstream[0].pvbusmaster	PVBusMaster

Name	Instance type
ARMCortexM7CT.l1_incoherent_interconnect.upstream[Z] (where Z = 0-17)	PVBusSlave
ARMCortexM7CT.l1dcache	PVCache
ARMCortexM7CT.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexM7CT.l1dcache.upstream[0]	PVBusSlave
ARMCortexM7CT.l1icache	PVCache
ARMCortexM7CT.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexM7CT.l1icache.upstream[0]	PVBusSlave
ARMCortexM7CT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

Name	Component type
ARMCortexM7CT	ARM_Cortex-M7
ARMCortexM7CT.acp_mapper	PVBusMapper
ARMCortexM7CT.ext_bus	PVBusLogger
ARMCortexM7CT.ext_bus.mapper	PVBusMapper
ARMCortexM7CT.l1_incoherent_interconnect	PVCache
ARMCortexM7CT.l1_incoherent_interconnect.downstream[0].pvbusmaster	PVBusMaster
ARMCortexM7CT.l1_incoherent_interconnect.upstream[Z] (where Z = 0-17)	PVBusSlave
ARMCortexM7CT.l1dcache	PVCache
ARMCortexM7CT.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexM7CT.l1dcache.upstream[0]	PVBusSlave
ARMCortexM7CT.l1icache	PVCache
ARMCortexM7CT.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexM7CT.l1icache.upstream[0]	PVBusSlave
ARMCortexM7CT.l2_flusher	AsyncCacheFlushUnit

## Ports for ARMCortexM7CT

Port	Direction	Protocol	Description
ahbd	slave	PVBus	Debug AHB - core bus slave driven by the DAP.
ahbp_m	master	PVBus	The core will generate Vendor System data accesses on this port.
ahbs	slave	PVBus	External master (e.g. DMA) can write TCMs (whether or not enabled in xTCMCR).
bigend	slave	Signal	Configure big endian data format.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
cpuwait	slave	Signal	When this signal is HIGH out of reset, it forces the processor into a quiescent state that delays its boot-up sequence and instruction execution until this signal is driven LOW
dap_s	slave	PVBus	Debug Access Port (DAP).
dbgen	slave	Signal	Invasive debug enable.
dbgrestart	slave	Signal	External debug request.
dbgrestarted	master	Signal	External debug request.

Port	Direction	Protocol	Description
edbgrq	slave	Signal	External debug request.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE and corresponds to the RTL TXEV and RXEV signals.
fpudisable	slave	Signal	Configure core with no FPU on reset.
fpxxc	master	Value	Port that sends the value of the FPxxC exception flags (FPIXC, FPIDC, FPOFC, FPUFC, FPDZC, FPIOC). Each flag is present in the same position as in the FPSCR register (e.g. fpxxc[0] = FPIOC).
halted	master	Signal	External debug request.
initahbpen	slave	Signal	Enable AHBP on the next reset
initvtor	slave	Value	Initial value of the Vector Table Offset Register (VTOR)
intisr	slave	Signal	This signal array delivers signals to the NVIC.
intnmi	slave	Signal	Configure non maskable interrupt.
lockup	master	Signal	Asserted when the processor is in lockup state.
mpudisable	slave	Signal	Configure core with no MPU on reset.
niden	slave	Signal	Non-invasive debug enable.
poreset	slave	Signal	Raising this signal will do a power-on reset of the core.
pv_ppbus_m	master	PVBus	The core will generate External Private Peripheral Bus requests on this port.
pvbus_m	master	PVBus	The core will generate bus requests on this port.
sleepdeep	master	Signal	Asserted when the processor is in deep sleep.
sleeping	master	Signal	Asserted when the processor is in sleep.
stcalib	slave	Value	This is the calibration value for the SysTick timer The following bit flag is modelled stcalib[25] - NOREF - Indicates no reference clock integrated. If stclk is not in use, set to 1
stclk	slave	ClockSignal	This is the reference clock for the SysTick timer, called STCLKEN in the specification
sysreset	slave	Signal	Raising this signal will put the core into reset mode (but does not reset the debug logic).
sysresetreq	master	Signal	Asserted to indicate that a reset is required.
ticks	master	InstructionCount	Port allowing the number of instructions since startup to be read from the CPU.

## Parameters for ARM CortexM7CT

### BIGENDINIT

Initialize processor to big endian mode.

Type: `bool`

Default value: `false`

### CFGAHBPSZ

Size of the AHBP port memory region. 0=AHBP disabled, 1=64MB, 2=128MB, 3=256MB, 4=512MB.

Type: `uint8_t`

Default value: 0



**CTI**

CTI (Cross Trigger Interface) included.

Type: `bool`

Default value: `false`

**CTI\_irq0\_pin**

CTI interrupt request 0 pin.

Type: `uint8_t`

Default value: 4

**CTI\_irq1\_pin**

CTI interrupt request 1 pin.

Type: `uint8_t`

Default value: 5

**DBG\_LVL**

0: 2 DWT, 4 FPB; 1: 4 DWT, 8 FPB comparators.

Type: `uint8_t`

Default value: 1

**DP\_FLOAT**

Support 8-byte floats.

Type: `bool`

Default value: `true`

**INIT\_AHBEN**

The AHB enable state at reset.

Type: `bool`

Default value: `false`

**INIT\_VTOR**

vector-table offset at reset.

Type: `uint32_t`

Default value: `0x0`

**LVL\_WIDTH**

Number of bits of interrupt priority.

Type: `uint8_t`

Default value: 3

**NUM\_IRQ**

Number of user interrupts.

Type: `uint8_t`

Default value: 32

**NUM\_MPU\_REGION**

Number of MPU regions.

Type: `uint8_t`

Default value: 16

**TRC**

Level of instrumentation trace supported. `false` : No ITM trace included, `true`: ITM trace included.

Type: `bool`

Default value: `true`

**WIC**

Include support for WIC-mode deep sleep.

Type: `bool`

Default value: `true`

**cpi\_div**

divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**cpi\_mul**

multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**dcache-size**

L1 D-cache size in bytes.

Type: `uint32_t`

Default value: `0x8000`

**dcache-state\_modelled**

Set whether D-cache has stateful implementation.

Type: `bool`

Default value: `false`

**dtcm\_enable**

Enable DTCM at reset.

Type: `bool`

Default value: `false`

**dtcm\_size**

DTCM size in KB.

Type: `uint16_t`

Default value: `0x100`

**icache-size**

L1 I-cache size in bytes.

Type: `uint32_t`

Default value: `0x8000`

**icache-state\_modelled**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

**itcm\_enable**

Enable ITCM at reset.

Type: `bool`

Default value: `false`

**itcm\_size**

ITCM size in KB.

Type: uint16\_t

Default value: 0x100

**manager\_id**

Manager ID presented in bus transactions.

Type: uint64\_t

Default value: 0x0

**min\_sync\_level**

force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**reported\_patch\_level**

Purely cosmetic patch level value to be displayed from the CUID register. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: int8\_t

Default value: -1

**reported\_revision\_number**

Purely cosmetic revision number value to be displayed from the CUID register. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: int8\_t

Default value: -1

**semihosting-Thumb\_SVC**

T32 SVC number for semihosting.

Type: uint8\_t

Default value: 171

**semihosting-cmd\_line**

Command line available to semihosting SVC calls.

Type: `string`

Default value: `N/A`

### **`semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

### **`semihosting-enable`**

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

Type: `bool`

Default value: `true`

### **`semihosting-heap_base`**

Virtual address of heap base.

Type: `uint32_t`

Default value: `0x0`

### **`semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint32_t`

Default value: `0x20700000`

### **`semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint32_t`

Default value: `0x20800000`

### **`semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint32_t`

Default value: `0x20700000`

**vfp-present**

Set whether the model has VFP support.

Type: `bool`

Default value: `true`

## 3.89 ARMCortexM85CT

Defined in `LISA/ARMCortexM85CT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r1p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following ports were removed:

- `auxfault`
- `currpri`

### About ARMCortexM85CT

This model supports the M-Profile Vector Extension (MVE) and the Custom Datapath Extension (CDE). For more information about CDE support in Fast Models, see [CDE](#).

### Iris and MTI instances for ARMCortexM85CT

This model has the following Iris instances:

Name	Instance type
ARMCortexM85CT	ARM_Cortex-M85
ARMCortexM85CT.acp_mapper	PVBusMapper
ARMCortexM85CT.ext_bus	PVBusLogger
ARMCortexM85CT.ext_bus.mapper	PVBusMapper
ARMCortexM85CT.l1_incoherent_interconnect	PVCache
ARMCortexM85CT.l1_incoherent_interconnect.downstream[0].pvbusmaster	PVBusMaster
ARMCortexM85CT.l1_incoherent_interconnect.upstream[Z] (where Z = 0-17)	PVBusSlave
ARMCortexM85CT.l1dcache	PVCache
ARMCortexM85CT.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexM85CT.l1dcache.upstream[0]	PVBusSlave
ARMCortexM85CT.l1licache	PVCache
ARMCortexM85CT.l1licache.downstream[0].pvbusmaster	PVBusMaster

Name	Instance type
ARMCortexM85CT.l1icache.upstream[0]	PVBusSlave
ARMCortexM85CT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

Name	Component type
ARMCortexM85CT	ARM_Cortex-M85
ARMCortexM85CT.acp_mapper	PVBusMapper
ARMCortexM85CT.ext_bus	PVBusLogger
ARMCortexM85CT.ext_bus.mapper	PVBusMapper
ARMCortexM85CT.l1_incoherent_interconnect	PVCache
ARMCortexM85CT.l1_incoherent_interconnect.downstream[0].pvbusmaster	PVBusMaster
ARMCortexM85CT.l1_incoherent_interconnect.upstream[Z] (where Z = 0-17)	PVBusSlave
ARMCortexM85CT.l1dcache	PVCache
ARMCortexM85CT.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexM85CT.l1dcache.upstream[0]	PVBusSlave
ARMCortexM85CT.l1icache	PVCache
ARMCortexM85CT.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexM85CT.l1icache.upstream[0]	PVBusSlave
ARMCortexM85CT.l2_flusher	AsyncCacheFlushUnit

### Ports for ARMCortexM85CT

Port	Direction	Protocol	Description
ahbd	slave	PVBus	Debug AHB - core bus slave driven by the DAP.
ahbp_m	master	PVBus	The core will generate Vendor System data accesses on this port.
ahbs	slave	PVBus	Slave AHB - External master (e.g. DMA) can write to TCMs (whether or not enabled in xTCMCR)
bigend	slave	Signal	Configure big endian data format.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions. Referred to in some documents as 'CLKIN'.
coproc_bus	slave	CoprocBusProtocol	Co-Processor Interface
core_dside_bus_gasket_in	slave	PVBus	Inbound D-side PVBus link this used when the external gasket is enabled. This allows externally processed D-side traffic to be fed back into the core path.
core_dside_bus_gasket_out	master	PVBus	Outbound D-side PVBus this exposed when has_core_dside_bus_gasket is true. This lets an external gasket observe or alter core data traffic before it re-enters on the inbound port.
cpuwait	slave	Signal	Stall the CPU out of reset
currns	master	Signal	Current Security state of the processor
dbgen	slave	Signal	Invasive debug enable
dbgrestart	slave	Signal	Request for synchronised exit from halt mode
dbgrestart	master	Signal	Handshakes with DBGRESTART

Port	Direction	Protocol	Description
edbgrq	slave	Signal	External request to enter halt mode
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE and corresponds to the RTL TXEV and RXEV signals.
fpxxc	master	Value	Port that sends the value of the FPxxC exception flags (FPIXC, FPIDC, FPOFC, FPUFC, FPDZC, FPIOC). Each flag is present in the same position as in the FPSCR register (e.g. fpxxc[0] = FPIOC).
halted	master	Signal	Indicates that the processor is in halt mode
idau_invalidate_region	slave	Value_64	64 bit number to invalid IDAU memory range (start_address<<32 end_address)
idau	master	PVBus	The core will generate IDAU requests on this port.
initnsvtor	slave	Value	Reset configuration port - Non-Secure Vector table offset (VTOR.TBLOFF[31:7]) out of reset This port remains functional no matter ARMv8-M Security Extensions are included or not When ARMv8-M Security Extensions are not included, all exceptions will use NS vector base address given by this port.
initpahben	slave	Signal	Enable P-AHB on the next reset
initsvtor	slave	Value	Reset configuration port - Secure Vector table offset (VTOR.TBLOFF[31:7]) out of reset It becomes functional when ARMv8-M Security Extensions are included When ARMv8-M Security Extensions are not included, this port will be ignored.
inittcm	slave	Signal	Reset configuration port - TCM enable initialisation out of reset Bit[0] HIGH = ITCM is enabled Bit[1] HIGH = DTCM is enabled
intnum	master	Value	Exception number of the current execution context (from IPSR[8:0]) When the processor is in Thread mode, INTNUM is 0 When the processor is in Handler mode, INTNUM is the exception number of the currently-executing exception.
irq	slave	Signal	This signal array delivers signals to the NVIC.
lockdcaic	slave	Signal	Disable access to instruction cache direct cache access registers
lockdtgu	slave	Signal	Disable writes to registers associated with the DTGU
lockitgu	slave	Signal	Disable writes to registers associated with the ITGU
locknsmpu	slave	Signal	Disable writes to the Non-Secure MPU_*_NS registers
locknsvtor	slave	Signal	Cortex-M85 specific LOCKNSVTOR, LOCKSVTAIRCR, LOCKSMPU, LOCKNSMPU, LOCKSAU. Disable writes to VTOR_NS
lockpahb	slave	Signal	P-AHB related ports Disable writes to PAHBCR
locksau	slave	Signal	Disable writes to the SAU_* registers
locksmpu	slave	Signal	Disable writes to the Secure MPU_* registers
locksvtaircr	slave	Signal	Disable writes to VTOR_S, AIRCR.PRIS, AIRCR.BFHFNMINS
locktcm	slave	Signal	Disable writes to registers associated with the TCM
lockup	master	Signal	Asserted when the processor is in lockup state.
niden	slave	Signal	Non-invasive debug enable
nmi	slave	Signal	Configure non maskable interrupt.
poreset	slave	Signal	Raising this signal will do a power-on reset of the core.
prequest	slave	PChannel	Low Power Interface
pv_ppbus_m	master	PVBus	The core will generate External Private Peripheral Bus requests on this port.
pdbus_m	master	PVBus	The core will generate bus requests on this port.



Port	Direction	Protocol	Description
grequest	slave	PChannel	Low Power Transition handshake interface
sleepdeep	master	Signal	Asserted when the processor is in deep sleep.
sleeping	master	Signal	Asserted when the processor is in sleep.
spiden	slave	Signal	Secure invasive debug enable
spniden	slave	Signal	Secure non-invasive debug enable
stcalib	slave	Value	These are the secure and non-secure calibration values for the SysTick timer, called CFGSSTCALIB/CFGNSSTCALIB in the M85 specification. stcalib[0] is the secure timer calibration, stcalib[1] is the non-secure timer calibration. In the case where the Security Extension is not present, stcalib[0] should be used. The following bit flag is modelled for each stcalib[*][25] - NOREF - Indicates no reference clock integrated. If stclk is not in use, set to 1
stclk	slave	ClockSignal	This is the reference clock for the SysTick timer, called SSTCLKEN/NSSTCLKEN in the M85 specification. This clock is provided for both secure and non secure inputs, even where ARMv8-M Security Extensions are included.
sysreset	slave	Signal	Raising this signal will put the core into reset mode (but does not reset the debug logic).
sysresetreq	master	Signal	Asserted to indicate that a reset is required.
ticks	master	InstructionCount	Port allowing the number of instructions since startup to be read from the CPU.
wicenack	master	Signal	Acknowledge signal for WICENREQ
wicenreq	slave	Signal	Request for deep sleep to be WIC-based deep sleep.
wicsense	master	Signal	Indicates which input events can cause the WIC to generate the WAKEUP signal.

## Parameters for ARM CortexM85CT

### CDEMAPPEDONCP

Bit N specifies whether the instruction for coprocessor N (CP7:CP0) is redirected to the CDE module.

Type: uint8\_t

Default value: 255

### CDERTLID

Value of ID\_AFR0.CDERTLID.

Type: uint8\_t

Default value: 32

### CFGBIGEND

Initialize processor to big endian mode.

Type: bool

Default value: `false`

### **CFGCPUINST**

CPU instance number. This is part of the TCM base address, in bits 25:24.

Type: `uint8_t`

Default value: 0

### **CFGDTCMSZ**

Size of the data TCM. 0=No DTCM implemented. Otherwise=Size of DTCM= $\text{pow}(2, \text{CFGDTCMSZ} - 1)$  KB. Minimum size is 4KB.

Type: `uint8_t`

Default value: 9

### **CFGITCMSZ**

Size of the instruction TCM. 0=No ITCM implemented. Otherwise=Size of ITCM= $\text{pow}(2, \text{CFGITCMSZ} - 1)$  KB. Minimum size is 4KB.

Type: `uint8_t`

Default value: 9

### **CFGMEMALIAS**

Memory address alias bit for the ITCM, DTCM and P-AHB regions. 0=No alias, 1=Alias bit 24, 2=Alias bit 25, 4=Alias bit 26, 8=Alias bit 27, 16=Alias bit 28.

Type: `uint8_t`

Default value: 0

### **CFGNOCDECP**

Bit N means external coprocessor N (CP7:CP0) disable for CDE coprocessor.

Type: `uint8_t`

Default value: 0

### **CFGPACBTI**

Enables support for the Pointer Authentication and Branch Target Identification (PACBTI) Extension. FALSE: Disabled, TRUE:PAC implemented using the QARMA3 algorithm with BTI.

Type: `bool`

Default value: `false`

**CFGPAHBSZ**

Size of the P-AHB peripheral port memory region. 0=P-AHB disabled, 1=64MB, 2=128MB, 3=256MB, 4=512MB.

Type: `uint8_t`

Default value: 0

**CPIF**

Specifies whether the external coprocessor interface is included.

Type: `bool`

Default value: `true`

**CPNSPRESENT**

Bit N means external coprocessor N (CP7:CP0) is accessible in Non-Secure state.

Type: `uint8_t`

Default value: 255

**CPSPPRESENT**

Bit N means external coprocessor N (CP7:CP0) is accessible in Secure state.

Type: `uint8_t`

Default value: 255

**CTI**

CTI (Cross Trigger Interface) included.

Type: `bool`

Default value: `false`

**CTI\_irq0\_pin**

CTI interrupt request 0 pin.

Type: `uint16_t`

Default value: 4

**CTI\_irq1\_pin**

CTI interrupt request 1 pin.

Type: `uint16_t`

Default value: 5

**DBGLVL**

1: 4 Watchpoints, 4 Breakpoint comparators; 2: 8 Watchpoints, 8 Breakpoint comparators.

Type: `uint8_t`

Default value: 2

**DCACHESZ**

Whether the D-cache is included and, if included, the size of it. Bit 0: 0=No D-cache included, 1=D-cache included. Bits [4:1]: 0x0=4KB D-cache, 0x1=8KB D-cache, 0x3=16KB D-cache, 0x7=32KB D-cache, 0xF=64KB D-cache.

Type: `uint8_t`

Default value: 15

**DTGU**

DTCM Security Gate Unit included.

Type: `bool`

Default value: `false`

**DTGUBLKSZ**

DTCM gate unit block size. Size= $\text{pow}(2, \text{DTGUBLKSZ} + 5)$  bytes.

Type: `uint8_t`

Default value: 3

**DTGUMAXBLKS**

Maximum number of DTCM gate unit blocks. Number of blocks= $\text{pow}(2, \text{DTGUMAXBLKS})$ .

Type: `uint8_t`

Default value: 0

**ECOREVNUM**

ECO Revision number.

Type: `uint64_t`

Default value: 0x0

**ERRDEVID.NUM**

RAS: Number of implemented error record indexes, 0 to 1.

Type: `uint8_t`

Default value: 1

**ETM**

Support for ETM trace. `false` : No ETM trace included, `true`: ETM trace included.

Type: `bool`

Default value: `true`

**FPU**

Set whether the model has VFP support.

Type: `bool`

Default value: `true`

**ICACHESZ**

Whether the I-cache is included and, if included, the size of it. Bit 0: 0=No I-cache included, 1=I-cache included. Bits [4:1]: `0x0`=4KB I-cache, `0x1`=8KB I-cache, `0x3`=16KB I-cache, `0x7`=32KB I-cache, `0xF`=64KB I-cache.

Type: `uint8_t`

Default value: 15

**INITNSVTOR**

Non-Secure vector-table offset at reset.

Type: `uint32_t`

Default value: `0x0`

**INITPAHBEN**

The P-AHB enable state at reset.

Type: `bool`

Default value: `false`

**INITSVTOR**

Secure vector-table offset at reset.

Type: `uint32_t`

Default value: `0x0`

### **INITTCMEN**

The TCM enable state at reset. Bit 0 corresponds to ITCM enable state, bit 1 corresponds to DTCM enable state.

Type: `uint8_t`

Default value: `3`

### **IRQDIS0**

IRQ line disable mask. Bit *n* of this 32-bit parameter disables `IRQ[n+0]`.

Type: `uint32_t`

Default value: `0x0`

### **IRQDIS1**

IRQ line disable mask. Bit *n* of this 32-bit parameter disables `IRQ[n+32]`.

Type: `uint32_t`

Default value: `0x0`

### **IRQDIS10**

IRQ line disable mask. Bit *n* of this 32-bit parameter disables `IRQ[n+320]`.

Type: `uint32_t`

Default value: `0x0`

### **IRQDIS11**

IRQ line disable mask. Bit *n* of this 32-bit parameter disables `IRQ[n+352]`.

Type: `uint32_t`

Default value: `0x0`

### **IRQDIS12**

IRQ line disable mask. Bit *n* of this 32-bit parameter disables `IRQ[n+384]`.

Type: `uint32_t`

Default value: `0x0`

**IRQDIS13**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+416].

Type: `uint32_t`

Default value: 0x0

**IRQDIS14**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+448].

Type: `uint32_t`

Default value: 0x0

**IRQDIS2**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+64].

Type: `uint32_t`

Default value: 0x0

**IRQDIS3**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+96].

Type: `uint32_t`

Default value: 0x0

**IRQDIS4**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+128].

Type: `uint32_t`

Default value: 0x0

**IRQDIS5**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+160].

Type: `uint32_t`

Default value: 0x0

**IRQDIS6**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+192].

Type: `uint32_t`

Default value: 0x0

**IRQDIS7**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+224].

Type: `uint32_t`

Default value: `0x0`

**IRQDIS8**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+256].

Type: `uint32_t`

Default value: `0x0`

**IRQDIS9**

IRQ line disable mask. Bit n of this 32-bit parameter disables IRQ[n+288].

Type: `uint32_t`

Default value: `0x0`

**IRQLVL**

Number of bits of interrupt priority.

Type: `uint8_t`

Default value: `3`

**ITGU**

ITCM Security Gate Unit included.

Type: `bool`

Default value: `false`

**ITGUBLKSZ**

ITCM gate unit block size. Size= $\text{pow}(2, \text{ITGUBLKSZ} + 5)$  bytes.

Type: `uint8_t`

Default value: `3`

**ITGUMAXBLKS**

Maximum number of ITCM gate unit blocks. Number of blocks= $\text{pow}(2, \text{ITGUMAXBLKS})$ .

Type: `uint8_t`

Default value: `0`



**ITM**

Level of instrumentation trace supported. `false` : No ITM trace included, `true`: ITM trace included.

Type: `bool`

Default value: `true`

**IWIC**

Include support for Internal Wake-up Interrupt Controller.

Type: `bool`

Default value: `true`

**LOCKDTGU**

Lock down of Data TGU registers write.

Type: `bool`

Default value: `false`

**LOCKITGU**

Lock down of Instruction TGU registers write.

Type: `bool`

Default value: `false`

**LOCKTCM**

Lock down of TCM registers write.

Type: `bool`

Default value: `false`

**LOCK\_NS\_MPU**

Lock down of Non-Secure MPU registers write.

Type: `bool`

Default value: `false`

**LOCK\_SAU**

Lock down of SAU registers write.

Type: `bool`

Default value: `false`

**LOCK\_S\_MPU**

Lock down of Secure MPU registers write.

Type: `bool`

Default value: `false`

**MPU\_NS**

Number of regions in the Non-Secure MPU. If Security Extensions are absent, this is the total number of MPU regions.

Type: `uint8_t`

Default value: 8

**MPU\_S**

Number of regions in the Secure MPU. If Security Extensions are absent, this is ignored.

Type: `uint8_t`

Default value: 8

**MVE**

Set whether the model has MVE support. If FPU = 0: 0=MVE not included, 1=Integer subset of MVE included. If FPU = 1: 0=MVE not included, 1=Integer subset of MVE included, 2=Integer and half and single precision floating point MVE included.

Type: `uint8_t`

Default value: 1

**NUMIRQ**

Number of user interrupts.

Type: `uint16_t`

Default value: 32

**SAU**

Number of SAU regions (0 => no SAU).

Type: `uint8_t`

Default value: 4

**SECEXT**

Whether the ARMv8-M Security Extensions are included.

Type: `bool`

Default value: `true`

### **WICLINES**

Number of lines supported by the WIC interface.

Type: `uint16_t`

Default value: `35`

### **cde\_impl\_name**

Name of the CDE implementation for this core (implementation contributed by MTI plugin).

Type: `string`

Default value: `N/A`

### **cpi\_div**

divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

### **cpi\_mul**

multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

### **dcache-state\_modelled**

Set whether D-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **delay\_faultmask\_update**

Delay FAULTMASK update to context sync.

Type: `bool`

Default value: `false`

### **delay\_sysreg\_update**

Delay some system register updates (e.g. SHCSR) to context sync.

Type: `bool`

Default value: `false`

### **`ecc_on`**

Enable Error Correcting Code.

Type: `bool`

Default value: `false`

### **`has_cde`**

Enables Custom Datapath Extensions.

Type: `bool`

Default value: `false`

### **`has_core_dside_bus_gasket`**

STL gasket enabled.

Type: `bool`

Default value: `false`

### **`icache-state_modelled`**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **`manager_id`**

Manager ID presented in bus transactions.

Type: `uint64_t`

Default value: `0x0`

### **`min_sync_level`**

force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

**reported\_patch\_level**

Purely cosmetic patch level value to be displayed from the CPUID register. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: `int8_t`

Default value: -1

**reported\_revision\_number**

Purely cosmetic revision number value to be displayed from the CPUID register. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: `int8_t`

Default value: -1

**semihosting-Thumb\_SVC**

T32 SVC number for semihosting.

Type: `uint8_t`

Default value: 171

**semihosting-cmd\_line**

Command line available to semihosting SVC calls.

Type: `string`

Default value: N/A

**semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**semihosting-enable**

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

Type: `bool`

Default value: `true`

**semihosting-heap\_base**

Virtual address of heap base.

Type: `uint32_t`

Default value: `0x0`

### **semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint32_t`

Default value: `0x20700000`

### **semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint32_t`

Default value: `0x20800000`

### **semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint32_t`

Default value: `0x20700000`

### **trace\_style**

MVE instruction trace style: Add 16 for `[*-]` beat trace. Add 32 for tracing IMPLIED LOB instructions. Add 64 to change opcode of implied BF to `0xBF00`.

Type: `uint8_t`

Default value: `2`

### **vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

## **3.90 ARMCortexR4CT**

Defined in `LISA/ARMCortexR4CT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r1p2	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

## About ARMCortexR4CT

- The model implements the `cfgie` port, although it is optional in hardware.
- `pvbuss` is the slave port to access the TCM RAM. Bits [3:0] of the user flags in the transaction are used to select the TCM:

**1**

selects the ATCM.

**2**

selects the BTCM.

Any other value is reserved.

- When you use the `semihosting-cmd_line` parameter to set the command line that is available to semihosting SVC calls, the value of `argv[0]` is set to the first command-line argument, not to the name of an image.

## Differences between the model and the RTL

This component has the following differences from the corresponding revision of the RTL implementation:

- There is a single memory port combining instruction, data, DMA and peripheral access.
- ECC and parity schemes are not supported (although the registers might be present).
- The dual core redundancy configuration is not supported.
- TCMs are modeled internally and the model does not support external TCMs or the ports associated with them.
- The hardware refers to the TCMs as “A” and “B”. The model refers to these as “i” and “d”.
- The RTL permits two data TCMs, B0 and B1, to be configured for extra bandwidth. These are not modeled.
- The `vfp-enable_at_reset` option is model-specific and has no hardware equivalent. Arm recommends that it is only used in test systems and tied off to false in production systems.

## Vectored Interrupt Controller (VIC) ports

The ARMCortexR4CT and ARMCortexR5xnCT models implement a simplified model of the Vectored Interrupt Controller (VIC) port.

The protocol consists of two ports:

- The `vic_addr` port signals the vectored interrupt address from the external VIC.
- The `vic_ack` port signals the VIC that an interrupt has been detected and is being serviced.

The expected interrupt sequence is:

1. The software enables the VIC interface by setting the VE bit in the CP15 control register and setting up suitable interrupt routines.
2. The VIC asserts IRQ.
3. Some time later, the processor detects and responds to the IRQ by asserting vic\_ack.
4. The VIC writes the vector address to the processor using vic\_addr.
5. The processor de-asserts vic\_ack.
6. The processor transfers control to the vectored address returned from the VIC.

The interaction between the processor and the VIC is untimed after the processor acknowledges the interrupt, so certain interrupt sequences cannot occur in the code translation processor models.

### Iris and MTI instances for ARM CortexR4CT

This model has the following Iris instances:

Name	Instance type
ARM CortexR4CT	ARM_Cortex-R4
ARM CortexR4CT.acp_mapper	PVBusMapper
ARM CortexR4CT.cpu0.l1dcache	PVCache
ARM CortexR4CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARM CortexR4CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARM CortexR4CT.cpu0.l1icache	PVCache
ARM CortexR4CT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARM CortexR4CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARM CortexR4CT.ext_bus	PVBusLogger
ARM CortexR4CT.ext_bus.mapper	PVBusMapper
ARM CortexR4CT.l1_incoherent_interconnect	PVCache
ARM CortexR4CT.l1_incoherent_interconnect.downstream[0].pvbusmaster	PVBusMaster
ARM CortexR4CT.l1_incoherent_interconnect.upstream[Z] (where Z = 0-17)	PVBusSlave
ARM CortexR4CT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

Name	Component type
ARM CortexR4CT	ARM_Cortex-R4
ARM CortexR4CT.acp_mapper	PVBusMapper
ARM CortexR4CT.cpu0.l1dcache	PVCache
ARM CortexR4CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARM CortexR4CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARM CortexR4CT.cpu0.l1icache	PVCache
ARM CortexR4CT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARM CortexR4CT.cpu0.l1icache.upstream[0]	PVBusSlave



Name	Component type
ARMCortexR4CT.ext_bus	PVBusLogger
ARMCortexR4CT.ext_bus.mapper	PVBusMapper
ARMCortexR4CT.l1_incoherent_interconnect	PVCache
ARMCortexR4CT.l1_incoherent_interconnect.downstream[0].pvbusmaster	PVBusMaster
ARMCortexR4CT.l1_incoherent_interconnect.upstream[Z] (where Z = 0-17)	PVBusSlave
ARMCortexR4CT.l2_flusher	AsyncCacheFlushUnit

## Ports for ARMCortexR4CT

Port	Direction	Protocol	Description
cfgend0	slave	Signal	Configure BE8 mode after a reset.
cfgie	slave	Signal	Configure big endian instruction format after a reset.
cfgnmfi	slave	Signal	Configure FIQs as non-maskable after a reset.
cfgte	slave	Signal	Configure exceptions to be taken in thumb mode after a reset.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
cpuhalt	slave	Signal	Raising this signal will put the core into halt mode.
fiq	slave	Signal	This signal drives the CPU's fast-interrupt handling.
initramd	slave	Signal	Configure DTCM enabled after a reset.
initrami	slave	Signal	Configure ITCM enabled after a reset.
irq	slave	Signal	This signal drives the CPU's interrupt handling.
loczrama	slave	Signal	Location of ATCM at reset.
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
pvbus_m	master	PVBus	The core will generate bus requests on this port.
pvbus_s	slave	PVBus	Slave access to TCMs.
reset	slave	Signal	Raising this signal will put the core into reset mode.
standbywfi	master	Signal	Signal from the core that it is waiting in standby for an interrupt.
ticks	master	InstructionCount	Port allowing the number of instructions since startup to be read from the CPU.
vic_ack	master	Signal	Vic acknowledge port to primary VIC.
vic_addr	slave	Value	Vic address port from primary VIC.
vinithi	slave	Signal	Configure high vectors after a reset.

## Parameters for ARMCortexR4CT

### CFGEND0

Initialize to BE8 endianness.

Type: bool

Default value: false

### CFGIE

Set the reset value of the instruction endian bit.

Type: `bool`

Default value: `false`

### **CFGNMF1**

Enable nonmaskable FIQ interrupts on startup.

Type: `bool`

Default value: `false`

### **CFGTE**

Initialize to take exceptions in T32 state. Model starts in T32 state.

Type: `bool`

Default value: `false`

### **INITRAMD**

Set or reset the INITRAMD signal.

Type: `bool`

Default value: `false`

### **INITRAMI**

Set or reset the INITRAMI signal.

Type: `bool`

Default value: `false`

### **LOCZRAMI**

Set or reset the LOCZRAMI signal.

Type: `bool`

Default value: `false`

### **NUM\_MPU\_REGION**

Number of MPU regions.

Type: `uint32_t`

Default value: `0x8`

### **VINITHI**

Initialize with high vectors enabled.

Type: `bool`

Default value: `false`

### **`cpi_div`**

divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: 1

### **`cpi_mul`**

multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: 1

### **`dcache-size`**

Set D-cache size in bytes.

Type: `uint32_t`

Default value: `0x10000`

### **`dcache-state_modelled`**

Set whether D-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **`dcm0_base`**

Base address of DTCM at startup.

Type: `uint32_t`

Default value: `0x800000`

### **`dcm0_size`**

Size of DTCM in KB.

Type: `uint32_t`

Default value: `0x8`

### **`icache-size`**

Set I-cache size in bytes.

Type: `uint32_t`

Default value: `0x10000`

### **`icache-state_modelled`**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **`implements_vfp`**

Set whether the model has been built with VFP support.

Type: `bool`

Default value: `true`

### **`itcm0_base`**

Base address of ITCM at startup.

Type: `uint32_t`

Default value: `0x0`

### **`itcm0_size`**

Size of ITCM in KB.

Type: `uint32_t`

Default value: `0x8`

### **`manager_id`**

Manager ID presented in bus transactions.

Type: `uint64_t`

Default value: `0x0`

### **`min_sync_level`**

force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint32_t`

Default value: `0`

**reported\_fp\_revision**

Purely cosmetic patch level value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored. Updates the FPSID.revision value.

Type: `int`

Default value: -1

**reported\_patch\_level**

Purely cosmetic patch level value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: `int`

Default value: -1

**reported\_revision\_number**

Purely cosmetic revision number value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: `int`

Default value: -1

**semihosting-ARM\_HLT**

ARM HLT number for semihosting.

Type: `uint32_t`

Default value: 0xF000

**semihosting-ARM\_SVC**

ARM SVC number for semihosting.

Type: `uint32_t`

Default value: 0x123456

**semihosting-Thumb\_HLT**

Thumb HLT number for semihosting.

Type: `uint32_t`

Default value: 0x3c

**semihosting-Thumb\_SVC**

Thumb SVC number for semihosting.

Type: `uint32_t`

Default value: `0xAB`

**semihosting-cmd\_line**

Command line available to semihosting SVC calls.

Type: `string`

Default value: `""`

**semihosting-cwd**

virtual address of CWD.

Type: `string`

Default value: `""`

**semihosting-enable**

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

Type: `bool`

Default value: `true`

**semihosting-heap\_base**

Virtual address of heap base.

Type: `uint32_t`

Default value: `0x0`

**semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint32_t`

Default value: `0xF000000`

**semihosting-hlt-enable**

Enable semihosting HLT traps. Applications that use HLT semihosting must set this parameter to true and the semihosting-enable parameter to true.

Type: `bool`

Default value: `false`

**semihosting-prefix**

Prefix semihosting output with target instance name.

Type: `bool`

Default value: `false`

**semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint32_t`

Default value: `0x10000000`

**semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint32_t`

Default value: `0xF000000`

**vfp-enable\_at\_reset**

Enable coprocessor access and VFP at reset.

Type: `bool`

Default value: `false`

3.91 ARMCortexR52CT

Defined in `LISA/ARMCortexR52CT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r1p4	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

Changes in 11.31.15

The following parameters were added:

- `has_delayed_dbgreg`

- `has_delayed_sysreg`

## About ARMCortexR52CT

The per-core parameters are preceded by `cpun.`, where `n` identifies the core (0-3).

- When you use the `semihosting-cmd_line` parameter to set the command line that is available to semihosting SVC calls, the value of `argv[0]` is set to the first command-line argument, not to the name of an image.
- The Cortex-R52 processor does not implement TrustZone technology, therefore the model does not support `S_*` or `NS_*` registers or exceptions.
- If flash memory is not enabled, to disable all routing to the flash port, set the `has_flash` parameter to `false`.

## Differences between the model and the RTL

- The model does not implement redundant cores for Dual-Core Lock-Step operations.
- The model does not implement the Low Power Interface to wake up the target core on receiving a `wake_request` signal from the GIC distributor.
- TCMs are modeled internally and the model does not support external TCMs or the ports associated with them.
- The model does not support running Software Test Libraries (STLs).
- The `vfp-enable_at_reset` parameter is a model-specific behavior with no hardware equivalent.
- ECC and parity schemes are hardware-specific so are not supported.

This model has a variable number of cores per cluster, specified using the `num_cores` parameter.

## Debug accesses

For debug accesses to succeed, they must have permissions that are compliant with the permission value in the `IMP_SLAVEPCTLR` register.



Note

The reset value of `IMP_SLAVEPCTLR` is `0x1` which means privileged access only.

## Iris and MTI instances for ARMCortexR52CT

This model has the following Iris instances:

Name	Instance type
ARMCortexR52CT	Cluster_ARM_Cortex-R52
ARMCortexR52CT.AMU	PVBusLogger
ARMCortexR52CT.AMU.mapper	PVBusMapper
ARMCortexR52CT.DAP	PVBusLogger
ARMCortexR52CT.DAP.mapper	PVBusMapper



Name	Instance type
ARMCortexR52CT.MMAP	PVBusLogger
ARMCortexR52CT.MMAP.mapper	PVBusMapper
ARMCortexR52CT.RAS	PVBusLogger
ARMCortexR52CT.RAS.mapper	PVBusMapper
ARMCortexR52CT.acp_mapper	PVBusMapper
ARMCortexR52CT.cpu0	ARM_CortexR52
ARMCortexR52CT.cpu0.UTLB	TLB
ARMCortexR52CT.cpu0.dtlb	TLB
ARMCortexR52CT.cpu0.gicv3_cpu_if	GICv3CPUInterface
ARMCortexR52CT.cpu0.l1dcache	PVCache
ARMCortexR52CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexR52CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexR52CT.cpu0.l1licache	PVCache
ARMCortexR52CT.cpu0.l1licache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexR52CT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexR52CT.ext_bus	PVBusLogger
ARMCortexR52CT.ext_bus.mapper	PVBusMapper
ARMCortexR52CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexR52CT.gic_iri	gic_iri
ARMCortexR52CT.gic_iri.rd_0	GICv3RedistributorInternal
ARMCortexR52CT.gic_iri.rd_0_0	GICv3RedistributorInternal
ARMCortexR52CT.gic_iri.rd_0_0_0	GICv3RedistributorInternal
ARMCortexR52CT.gic_iri.rd_0_0_0_v (where V = 0-1)	GICv3Redistributor
ARMCortexR52CT.gic_iri.rd_0_0_0_v.ExportTest.ARM_CortexR52CT.gic_iri.rd_0_0_0_D.pvbus_m[0].pvbusmaster (where V = 0-1; D = 0-1)	PVBusMaster
ARMCortexR52CT.gic_iri.rd_tl	GICv3Distributor
ARMCortexR52CT.global_debug_rom	debug_rom
ARMCortexR52CT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

Name	Component type
ARMCortexR52CT.AMU	PVBusLogger
ARMCortexR52CT.AMU.mapper	PVBusMapper
ARMCortexR52CT.DAP	PVBusLogger
ARMCortexR52CT.DAP.mapper	PVBusMapper
ARMCortexR52CT.MMAP	PVBusLogger
ARMCortexR52CT.MMAP.mapper	PVBusMapper
ARMCortexR52CT.RAS	PVBusLogger
ARMCortexR52CT.RAS.mapper	PVBusMapper
ARMCortexR52CT.acp_mapper	PVBusMapper

Name	Component type
ARMCortexR52CT.cpu0	ARM_CortexR52
ARMCortexR52CT.cpu0.UTLB	TLB
ARMCortexR52CT.cpu0.gicv3_cpu_if	GICv3CPUInterface
ARMCortexR52CT.cpu0.l1dcache	PVCache
ARMCortexR52CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexR52CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexR52CT.cpu0.l1licache	PVCache
ARMCortexR52CT.cpu0.l1licache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexR52CT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexR52CT.ext_bus	PVBusLogger
ARMCortexR52CT.ext_bus.mapper	PVBusMapper
ARMCortexR52CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexR52CT.gic_iri.rd_0	GICv3RedistributorInternal
ARMCortexR52CT.gic_iri.rd_0_0	GICv3RedistributorInternal
ARMCortexR52CT.gic_iri.rd_0_0_0	GICv3RedistributorInternal
ARMCortexR52CT.gic_iri.rd_0_0_0_V (where V = 0-1)	GICv3Redistributor
ARMCortexR52CT.gic_iri.rd_0_0_0_V.ExportTest.ARM_CortexR52CT.gic_iri.rd_0_0_0_D.pvbus_m[0].pvbusmaster (where V = 0-1; D = 0-1)	PVBusMaster
ARMCortexR52CT.gic_iri.rd_tl	GICv3Distributor
ARMCortexR52CT.l2_flusher	AsyncCacheFlushUnit

## Ports for ARMCortexR52CT

Port	Direction	Protocol	Description
cfgdbgromaddr	slave	Value_64	Debug ROM base address.
cfgdbgromaddrv	slave	Signal	Debug ROM base address valid.
cfgendianess	slave	Signal	This signal if for EE bit initialisation.
cfgperiphbase	slave	Value_64	This port sets the base address of private peripheral region.
cfgthumbexceptions	slave	Signal	This signal provides default exception handling state.
cfgvectable	slave	Value_64	Reset vector base address.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
clrexmonack	master	Signal	Acknowledge handshake signal for the clrexmonreq signal
clrexmonreq	slave	Signal	Signals the clearing of an external global exclusive monitor
clusterid	slave	Value	Configure Aff2 and Aff1 fields of MPIDR. Aff1 = value[7:0], Aff2 = value[15:8]
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.
commr	master	Signal	Receive portion of Data Transfer Register full.
commtx	master	Signal	Transmit portion of Data Transfer Register empty.

Port	Direction	Protocol	Description
COREPACTIVEx1	master	Signal	These signals relate to core power down. Equivalent to COREPACTIVEx[1]
cpuhalt	slave	Signal	Raising this signal will put the core into halt mode.
cpuporeset	slave	Signal	Power on reset. Initializes all the processor logic, including debug logic.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
dbgack	master	Signal	External debug interface.
dbgen	slave	Signal	External debug interface.
dbgnopwrdown	master	Signal	There is no support for PChannel in CortexR52. These signals relate to core power down. Equivalent to COREPACTIVEx[0]
dbgpwrupreq	master	Signal	Debug power up request.
dev_debug_s	slave	PVBus	External debug interface.
edbgrq	slave	Signal	External debug interface.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
ext_slave_s	slave	PVBus	External Slave port. Equivalent to AXIS port
extppi_in_0	slave	Signal	Core 0 external ppi signals.
extppi_in_1	slave	Signal	Core 1 external ppi signals.
extppi_in_2	slave	Signal	Core 2 external ppi signals.
extppi_in_3	slave	Signal	Core 3 external ppi signals.
flash_m	master	PVBus	Flash Port.
gdu_external_m	master	GICv3Comms	GDU external messaging port.
hiden	slave	Signal	External debug interface.
hniden	slave	Signal	External debug interface.
llpp_m	master	PVBus	LLPP (Low-Latency Peripheral Port).
memorymapped_debug_s	slave	PVBus	External debug interface.
niden	slave	Signal	External debug interface.
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_core_m	master	PVBus	The core will generate bus requests on this port. Equivalent to AXIM port
reset	slave	Signal	Raising this signal will put the core into reset mode.
sei	slave	Signal	Per core virtual System Error physical pins.
spi_in	slave	Signal	Shared peripheral interrupts.
standbywfe	master	Signal	This signal indicates if a core is in WFE state.
standbywfi	master	Signal	This signal indicates if a core is in WFI state.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
topreset	slave	Signal	This signal resets timer and interrupt controller.
vsei	slave	Signal	Per core virtual System Error physical pins.
warmrstreq	master	Signal	Warm reset request from core.

## Parameters for ARMCortexR52CT

### **cpuX.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian.

Type: `bool`

Default value: `false`

### **cpuX.CFGTE**

Equivalent to CFGTHUMBEXCEPTIONS.

Type: `bool`

Default value: `false`

### **cpuX.RVBARADDR**

Equivalent to CFGVECTABLE.

Type: `uint32_t`

Default value: `0x0`

### **cpuX.ase-present**

Set whether the model has been built with NEON support.

Type: `bool`

Default value: `true`

### **cpuX.dcache-size**

L1 D-Cache size in bytes.

Type: `uint16_t`

Default value: `0x8000`

### **cpuX.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **cpuX.flash.enable**

Equivalent to CFGFLASHEN.

Type: `bool`

Default value: `false`

### **`cpuX.icache-size`**

L1 I-Cache size in bytes.

Type: `uint16_t`

Default value: `0x8000`

### **`cpuX.llpp.base`**

Equivalent to `CFGLLPPBASEADDR`.

Type: `uint32_t`

Default value: `0x0`

### **`cpuX.llpp.size`**

Equivalent to `CFGLLPPSIZE`.

Type: `uint32_t`

Default value: `0x1000`

### **`cpuX.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint64_t`

Default value: `0x4000000`

### **`cpuX.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

### **`cpuX.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**cpuX.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**cpuX.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**cpuX.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**cpuX.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**cpuX.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**cpuX.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**cpuX.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint32_t`

Default value: `0x0`

**cpuX.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint32_t`

Default value: `0xf000000`

**cpuX.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint32_t`

Default value: `0x10000000`

**cpuX.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint32_t`

Default value: `0xf000000`

**cpuX.tcm.a.enable**

Equivalent to CFGTCMBOOT.

Type: `bool`

Default value: `false`

**cpuX.tcm.a.size**

Sets the size of the ATCM(in bytes).

Type: `uint32_t`

Default value: `0x4000`

**cpuX.tcm.a.wait**

TCM Register A accesses wait states: 0-1 states.

Type: `uint32_t`

Default value: `0x0`

**cpuX.tcm.b.size**

Sets the size of the BTCM(in bytes).

Type: `uint32_t`

Default value: `0x4000`

**cpuX.tcm.b.wait**

TCM Register B accesses wait states: 0-1 states.

Type: uint32\_t

Default value: 0x0

**cpuX.tcm.c.size**

Sets the size of the CTCM(in bytes).

Type: uint32\_t

Default value: 0x2000

**cpuX.tcm.c.wait**

TCM Register C accesses wait states: 0-1 states.

Type: uint32\_t

Default value: 0x0

**cpuX.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**cpuX.vfp-dp-present**

Whether double-precision floating point feature is implemented.

Type: bool

Default value: true

**cpuX.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**CLUSTER\_ID**

CLUSTER\_ID[15:8] equivalent to CFGMPIDRAFF2, CLUSTER\_ID[7:0] equivalent to CFGMPIDRAFF1.



Type: `uint16_t`

Default value: `0x0`

### **CMO\_broadcast\_when\_cache\_state\_modelling\_disabled**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. This value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required.

Type: `uint8_t`

Default value: `1`

### **DBGROMADDR**

Equivalent to `CFGDBGROMADDR`.

Type: `uint32_t`

Default value: `0x0`

### **DBGROMADDRV**

If true, set bits[1:0] of the CP15 `DBGDRAR` to indicate that the address is valid.

Type: `bool`

Default value: `false`

### **GICDISABLE**

Used internally. Please ignore.

Type: `bool`

Default value: `false`

### **NUM\_CORES**

Number of cores in cluster.

Type: `uint8_t`

Default value: `1`

### **PERIPHBASE**

Equivalent to `CFGPERIPHBASE`.

Type: `uint32_t`

Default value: 0x13080000

**cluster\_utid**

Equivalent to CFGCLUSTERUTID.

Type: uint8\_t

Default value: 0

**cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: uint32\_t

Default value: 0x1

**cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: uint32\_t

Default value: 0x1

**dcache-prefetch\_enabled**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

Type: bool

Default value: false

**dcache-read\_access\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**dcache-state\_modelled**

Set whether D-cache has stateful implementation.

Type: bool

Default value: false

**dcache-write\_access\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**enable\_lock\_step**

(equivalent to `CFGSLSPPLIT`).

Type: `bool`

Default value: `false`

**enable\_simulation\_performance\_optimizations**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of `stage12_tlb_size` parameter to 1024).

Type: `bool`

Default value: `true`

**flash\_protection\_enable\_at\_reset**

Equivalent to `CFGFLASHPROTEN`.

Type: `bool`

Default value: `false`

**has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**has\_export\_m\_port**

The interrupt distributor has an optional interrupt export port for routing interrupts to an external device.

Type: `bool`

Default value: `true`

**has\_flash**

Equivalent to `CFGFLASHIMP`. `has_flash = false => CFGFLASHIMP = false`.

Type: `bool`

Default value: `false`

**has\_flash\_protection**

Equivalent to `CFGFLASHPROTIMP`.

Type: `bool`

Default value: `true`

**has\_llpp**

Equivalent to `CFGLLPPIMP`.

Type: `bool`

Default value: `false`

**icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

**icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-state\_modelled**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

**memory.ext\_slave\_base**

Equivalent to `CFGAXISTCMBASEADDR`.

Type: `uint32_t`

Default value: `0x0`

**memory.flash\_base**

Equivalent to `CFGFLASHBASEADDR`.

Type: `uint32_t`

Default value: `0x0`

**num\_protection\_regions\_s1**

Number of v8-R stage1 protection regions.

Type: `uint8_t`

Default value: `24`

**num\_protection\_regions\_s2**

Number of v8-R hyp protection regions.

Type: `uint8_t`

Default value: `16`

**num\_spi**

Number of interrupts (SPI) into the internal GIC controller.

Type: `uint16_t`

Default value: `960`

**ram\_protection\_enable\_at\_reset**

Equivalent to `CFGGRAMPROTEN`.

Type: `bool`

Default value: `false`

**reported\_fp\_revision**

Purely cosmetic patch level value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored. Updates the FPSID.revision value.

Type: `int`

Default value: -1

**reported\_patch\_level**

Purely cosmetic patch level value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: `int`

Default value: -1

**reported\_revision\_number**

Purely cosmetic revision number value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: `int`

Default value: -1

## 3.92 ARMCortexR52PlusCT

Defined in `LISA/ARMCortexR52PlusCT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following parameters were added:

- `has_delayed_dbgreg`
- `has_delayed_sysreg`

### About ARMCortexR52PlusCT

The model does not implement the following:

- Redundant cores for Dual Core Lock Step operation.

- Low Power Interface to wake the target core on receiving a `wake_request` from the GIC Distributor.

## Debug accesses

For debug accesses to succeed, they must have permissions that are compliant with the permission value in the `IMP_SLAVEPCTLR` register.



The reset value of `IMP_SLAVEPCTLR` is `0x1` which means privileged access only.

## Iris and MTI instances for ARMCortexR52PlusCT

This model has the following Iris instances:

Name	Instance type
ARMCortexR52PlusCT	Cluster_ARM_Cortex-R52Plus
ARMCortexR52PlusCT.AMU	PVBusLogger
ARMCortexR52PlusCT.AMU.mapper	PVBusMapper
ARMCortexR52PlusCT.DAP	PVBusLogger
ARMCortexR52PlusCT.DAP.mapper	PVBusMapper
ARMCortexR52PlusCT.MMAP	PVBusLogger
ARMCortexR52PlusCT.MMAP.mapper	PVBusMapper
ARMCortexR52PlusCT.RAS	PVBusLogger
ARMCortexR52PlusCT.RAS.mapper	PVBusMapper
ARMCortexR52PlusCT.acp_mapper	PVBusMapper
ARMCortexR52PlusCT.cpu0	ARM_Cortex-R52Plus
ARMCortexR52PlusCT.cpu0.UTLB	TLB
ARMCortexR52PlusCT.cpu0.dtlb	TLB
ARMCortexR52PlusCT.cpu0.gicv3_cpu_if	GICv3CPUInterface
ARMCortexR52PlusCT.cpu0.l1dcache	PVCache
ARMCortexR52PlusCT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexR52PlusCT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexR52PlusCT.cpu0.l1icache	PVCache
ARMCortexR52PlusCT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexR52PlusCT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexR52PlusCT.ext_bus	PVBusLogger
ARMCortexR52PlusCT.ext_bus.mapper	PVBusMapper
ARMCortexR52PlusCT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexR52PlusCT.gic_iri	gic_iri
ARMCortexR52PlusCT.gic_iri.rd_0	GICv3RedistributorInternal
ARMCortexR52PlusCT.gic_iri.rd_0_0	GICv3RedistributorInternal

Name	Instance type
ARMCortexR52PlusCT.gic_iri.rd_0_0_0	GlCv3RedistributorInternal
ARMCortexR52PlusCT.gic_iri.rd_0_0_0_V (where V = 0-1)	GlCv3Redistributor
ARMCortexR52PlusCT.gic_iri.rd_0_0_0_V.ExportTest.ARM CortexR52PlusCT.gic_iri.rd_0_0_0_D.pvbus_m[0].pvbusmaster (where V = 0-1; D = 0-1)	PVBusMaster
ARMCortexR52PlusCT.gic_iri.rd_t1	GlCv3Distributor
ARMCortexR52PlusCT.global_debug_rom	debug_rom
ARMCortexR52PlusCT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

Name	Component type
ARMCortexR52PlusCT.AMU	PVBusLogger
ARMCortexR52PlusCT.AMU.mapper	PVBusMapper
ARMCortexR52PlusCT.DAP	PVBusLogger
ARMCortexR52PlusCT.DAP.mapper	PVBusMapper
ARMCortexR52PlusCT.MMAP	PVBusLogger
ARMCortexR52PlusCT.MMAP.mapper	PVBusMapper
ARMCortexR52PlusCT.RAS	PVBusLogger
ARMCortexR52PlusCT.RAS.mapper	PVBusMapper
ARMCortexR52PlusCT.acp_mapper	PVBusMapper
ARMCortexR52PlusCT.cpu0	ARM_Cortex-R52Plus
ARMCortexR52PlusCT.cpu0.UTLB	TLB
ARMCortexR52PlusCT.cpu0.gicv3_cpu_if	GlCv3CPUInterface
ARMCortexR52PlusCT.cpu0.l1dcache	PVCache
ARMCortexR52PlusCT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexR52PlusCT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexR52PlusCT.cpu0.l1icache	PVCache
ARMCortexR52PlusCT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexR52PlusCT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexR52PlusCT.ext_bus	PVBusLogger
ARMCortexR52PlusCT.ext_bus.mapper	PVBusMapper
ARMCortexR52PlusCT.gic_cpuif_decoder_cluster	GlCv3CPUInterfaceDecoder
ARMCortexR52PlusCT.gic_iri.rd_0	GlCv3RedistributorInternal
ARMCortexR52PlusCT.gic_iri.rd_0_0	GlCv3RedistributorInternal
ARMCortexR52PlusCT.gic_iri.rd_0_0_0	GlCv3RedistributorInternal
ARMCortexR52PlusCT.gic_iri.rd_0_0_0_V (where V = 0-1)	GlCv3Redistributor
ARMCortexR52PlusCT.gic_iri.rd_0_0_0_V.ExportTest.ARM CortexR52PlusCT.gic_iri.rd_0_0_0_D.pvbus_m[0].pvbusmaster (where V = 0-1; D = 0-1)	PVBusMaster
ARMCortexR52PlusCT.gic_iri.rd_t1	GlCv3Distributor
ARMCortexR52PlusCT.l2_flusher	AsyncCacheFlushUnit



## Ports for ARMCortexR52PlusCT

Port	Direction	Protocol	Description
cfgdbgromaddr	slave	Value_64	Debug ROM base address.
cfgdbgromaddrv	slave	Signal	Debug ROM base address valid.
cfgendianess	slave	Signal	This signal if for EE bit initialisation.
cfgperiphbase	slave	Value_64	This port sets the base address of private peripheral region.
cfgthumbexceptions	slave	Signal	This signal provides default exception handling state.
cfgvectable	slave	Value_64	Reset vector base address.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
clusterid	slave	Value	Configure Aff2 and Aff1 fields of MPIDR. Aff1 = value[7:0], Aff2 = value[15:8]
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.
commr	master	Signal	Receive portion of Data Transfer Register full.
commt	master	Signal	Transmit portion of Data Transfer Register empty.
cpuhalt	slave	Signal	Raising this signal will put the core into halt mode.
cpuporeset	slave	Signal	Power on reset. Initializes all the processor logic, including debug logic.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
dbgack	master	Signal	External debug interface.
dbgen	slave	Signal	External debug interface.
dbgnopwrdown	master	Signal	There is no support for PChannel in CortexR52Plus. These signals relate to core power down. Equivalent to COREPACTIVEx
dbgpwrupreq	master	Signal	Debug power up request.
dev_debug_s	slave	PVBus	External debug interface.
edbgrq	slave	Signal	External debug interface.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
ext_slave_s	slave	PVBus	External Slave port. Equivalent to AXIS port
extppi_in_0	slave	Signal	Core 0 external ppi signals.
extppi_in_1	slave	Signal	Core 1 external ppi signals.
extppi_in_2	slave	Signal	Core 2 external ppi signals.
extppi_in_3	slave	Signal	Core 3 external ppi signals.
flash_m	master	PVBus	Flash Port.
gdu_external_m	master	GLCv3Comms	GDU external messaging port.
hiden	slave	Signal	External debug interface.
hniden	slave	Signal	External debug interface.
llpp_m	master	PVBus	LLPP (Low-Latency Peripheral Port).
memorymapped_debug_s	slave	PVBus	External debug interface.
niden	slave	Signal	External debug interface.

Port	Direction	Protocol	Description
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_core_m	master	PVBus	The core will generate bus requests on this port. Equivalent to AXIM port
reset	slave	Signal	Raising this signal will put the core into reset mode.
sei	slave	Signal	Per core virtual System Error physical pins.
spi_in	slave	Signal	Shared peripheral interrupts.
standbywfe	master	Signal	This signal indicates if a core is in WFE state.
standbywfi	master	Signal	This signal indicates if a core is in WFI state.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
topreset	slave	Signal	This signal resets timer and interrupt controller.
vsei	slave	Signal	Per core virtual System Error physical pins.
warmrstreq	master	Signal	Warm reset request from core.

## Parameters for ARM Cortex R52 Plus CT

### **cpuX.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **cpuX.CFGTE**

Equivalent to CFGTHUMBEXCEPTIONS.

Type: `bool`

Default value: `false`

### **cpuX.RVBARADDR**

Equivalent to CFGVECTABLE.

Type: `uint32_t`

Default value: `0x0`

### **cpuX.ase-present**

Set whether the model has been built with NEON support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**cpuX.dcache-size**

L1 D-Cache size in bytes.

Type: `uint16_t`

Default value: `0x8000`

**cpuX.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**cpuX.flash.enable**

Equivalent to `CFGFLASHEN`.

Type: `bool`

Default value: `false`

**cpuX.icache-size**

L1 I-Cache size in bytes.

Type: `uint16_t`

Default value: `0x8000`

**cpuX.llpp.base**

Equivalent to `CFGLLPPBASEADDR`.

Type: `uint64_t`

Default value: `0x0`

**cpuX.llpp.size**

Equivalent to `CFGLLPPSIZE`.

Type: `uint32_t`

Default value: `0x8000000`

**cpuX.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

### **`cpuX.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

### **`cpuX.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`cpuX.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

### **`cpuX.semihosting-T32_HLT`**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

### **`cpuX.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

### **`cpuX.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

### **`cpuX.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**`cpuX.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**`cpuX.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint32_t`

Default value: `0x0`

**`cpuX.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint32_t`

Default value: `0xf000000`

**`cpuX.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint32_t`

Default value: `0x10000000`

**`cpuX.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint32_t`

Default value: `0xf000000`

**`cpuX.tcm.a.enable`**

Equivalent to CFGTCMBOOT.

Type: `bool`

Default value: `false`

**`cpuX.tcm.a.size`**

Sets the size of the ATCM(in bytes).

Type: `uint32_t`

Default value: `0x4000`

#### **`cpuX.tcm.a.wait`**

TCM Register A accesses wait states: 0-1 states.

Type: `uint8_t`

Default value: `0`

#### **`cpuX.tcm.b.size`**

Sets the size of the BTCM(in bytes).

Type: `uint32_t`

Default value: `0x4000`

#### **`cpuX.tcm.b.wait`**

TCM Register B accesses wait states: 0-1 states.

Type: `uint8_t`

Default value: `0`

#### **`cpuX.tcm.c.size`**

Sets the size of the CTCM(in bytes).

Type: `uint32_t`

Default value: `0x2000`

#### **`cpuX.tcm.c.wait`**

TCM Register C accesses wait states: 0-1 states.

Type: `uint8_t`

Default value: `0`

#### **`cpuX.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**cpuX.vfp-dp-present**

Whether double-precision floating point feature is implemented (FEAT\_F64MM).

Type: `bool`

Default value: `true`

**cpuX.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**CLUSTER\_ID**

CLUSTER\_ID[15:8] equivalent to CFGMPIDRAFF2, CLUSTER\_ID[7:0] equivalent to CFGMPIDRAFF1.

Type: `uint16_t`

Default value: `0x0`

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain even when cache state modelling is disabled. Possible values are:- 0 = All CMOs are broadcast if architecturally required- 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required.

Type: `uint8_t`

Default value: `1`

**DBGROMADDR**

Equivalent to CFGDBGROMADDR.

Type: `uint32_t`

Default value: `0x0`

**DBGROMADDRV**

If true, set bits[1:0] of the CP15 DBGDRAR to indicate that the address is valid.

Type: `bool`

Default value: `false`

**GICDISABLE**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`

Default value: `false`

**NUM\_CORES**

Number of cores in cluster.

Type: `uint8_t`

Default value: 1

**PERIPHBASE**

Equivalent to CFGPERIPHBASE.

Type: `uint32_t`

Default value: `0x13080000`

**cluster\_utid**

Equivalent to CFGCLUSTERUTID.

Type: `uint8_t`

Default value: 0

**cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**dcache-prefetch\_enabled**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

Type: `bool`



Default value: `false`

### **`dcache-read_access_latency`**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`dcache-state_modelled`**

Set whether D-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **`dcache-write_access_latency`**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`enable_lock_step`**

(equivalent to `CFGSLSPPLIT`).

Type: `bool`

Default value: `false`

### **`enable_simulation_performance_optimizations`**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of `stage12_tlb_size` parameter to 1024).

Type: `bool`

Default value: `true`

**flash\_protection\_enable\_at\_reset**

Equivalent to CFGFLASHPROTEN.

Type: `bool`

Default value: `false`

**has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**has\_export\_m\_port**

The interrupt distributor has an optional interrupt export port for routing interrupts to an external device.

Type: `bool`

Default value: `true`

**has\_flash**

Equivalent to CFGFLASHIMP.

Type: `bool`

Default value: `false`

**has\_flash\_protection**

Equivalent to CFGFLASHPROTIMP.

Type: `bool`

Default value: `true`

**has\_llpp**

Equivalent to CFGLLPPIMP.

Type: `bool`

Default value: `false`

### **icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

### **icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **icache-state\_modelled**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **memory.ext\_slave\_base**

Equivalent to `CFGAXISTCMBASEADDR`.

Type: `uint64_t`

Default value: `0x0`

### **memory.flash\_base**

Equivalent to `CFGFLASHBASEADDR`.

Type: `uint32_t`

Default value: `0x0`

### **num\_protection\_regions\_s1**

Number of v8-R protection regions.

Type: `uint8_t`

Default value: `24`

**num\_protection\_regions\_s2**

Number of v8-R hyp protection regions.

Type: `uint8_t`

Default value: 24

**num\_spi**

Number of interrupts (SPI) into the internal GIC controller.

Type: `uint16_t`

Default value: 960

**ram\_protection\_enable\_at\_reset**

Equivalent to CFGRAMPROTEN.

Type: `bool`

Default value: `false`

**reported\_fp\_revision**

Purely cosmetic patch level value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored. Updates the FPSID.revision value.

Type: `int8_t`

Default value: -1

**reported\_patch\_level**

Purely cosmetic patch level value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: `int8_t`

Default value: -1

**reported\_revision\_number**

Purely cosmetic revision number value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: `int8_t`

Default value: -1

## 3.93 ARMCortexR5x1CT

Defined in `LISA/ARMCortexR5x1CT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r1p2	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### About ARMCortexR5x1CT

An ARMCortexR5x2CT component also exists.

The per-core parameters are preceded by `cpun.`, where `n` identifies the core (0 or 1).

The allowed values for the `LOCK_STEP` parameter are:

- 0**  
Disable. Set for two independent cores.
- 1**  
Lock Step. Appears to the system as two cores but is internally modeled as a single core.
- 3**  
Split Lock. Appears to the system as two cores but can be statically configured from reset either as two independent cores or two locked cores. For the model, these are equivalent to Disable and Lock Step, respectively, except for the value of build options registers. The model does not support dynamically splitting and locking the cluster.

`pvbus_s` is the slave port to access the TCM RAM of CPU `n`. Bits [3:0] of the user flags in the transaction are used to select the TCM:

- 1**  
Selects the ATCM of CPU 0
- 2**  
Selects the BTCM of CPU 0
- 3**  
Selects the ATCM of CPU 1
- 4**  
Selects the BTCM of CPU 1

Any other value is reserved.

### Differences between the model and the RTL

This component has the following differences from the corresponding revision of the RTL implementation:

- The RR bit in the SCTLR is ignored.
- The Low Latency Peripheral Port is not modeled.
- The model only has a single bus master port combining instruction, data, DMA and peripheral accesses. The CP15 control registers associated with peripheral buses preserve values but do not have any other effect.
- The model only supports static split lock and not dynamic split lock. Contact Arm for details.
- TCMs are modeled internally and the model does not support external TCMs or the ports associated with them.
- The model cannot experience an ECC error and does not support fault injection into the system, so Arm does not provide the ability to set error schemes for the caches or TCMs. Contact Arm if you require a particular value in the Build Options registers.

### Vectored Interrupt Controller (VIC) ports

The ARMCortexR4CT and ARMCortexR5xnCT models implement a simplified model of the Vectored Interrupt Controller (VIC) port.

The protocol consists of two ports:

- The `vic_addr` port signals the vectored interrupt address from the external VIC.
- The `vic_ack` port signals the VIC that an interrupt has been detected and is being serviced.

The expected interrupt sequence is:

1. The software enables the VIC interface by setting the VE bit in the CP15 control register and setting up suitable interrupt routines.
2. The VIC asserts IRQ.
3. Some time later, the processor detects and responds to the IRQ by asserting `vic_ack`.
4. The VIC writes the vector address to the processor using `vic_addr`.
5. The processor de-asserts `vic_ack`.
6. The processor transfers control to the vectored address returned from the VIC.

The interaction between the processor and the VIC is untimed after the processor acknowledges the interrupt, so certain interrupt sequences cannot occur in the code translation processor models.

### Iris and MTI instances for ARMCortexR5x1CT

This model has the following Iris instances:

Name	Instance type
ARMCortexR5x1CT	Cluster_ARM_Cortex-R5
ARMCortexR5x1CT.acp_mapper	PVBusMapper
ARMCortexR5x1CT.cpu0	ARM_Cortex-R5
ARMCortexR5x1CT.cpu0.l1dcache	PVCache
ARMCortexR5x1CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster

Name	Instance type
ARMCortexR5x1CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexR5x1CT.cpu0.l1icache	PVCache
ARMCortexR5x1CT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexR5x1CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexR5x1CT.ext_bus	PVBusLogger
ARMCortexR5x1CT.ext_bus.mapper	PVBusMapper
ARMCortexR5x1CT.l1_incoherent_interconnect	PVCache
ARMCortexR5x1CT.l1_incoherent_interconnect.downstream[0].pvbusmaster	PVBusMaster
ARMCortexR5x1CT.l1_incoherent_interconnect.upstream[U] (where U = 0-17)	PVBusSlave
ARMCortexR5x1CT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

Name	Component type
ARMCortexR5x1CT.acp_mapper	PVBusMapper
ARMCortexR5x1CT.cpu0	ARM_Cortex-R5
ARMCortexR5x1CT.cpu0.l1dcache	PVCache
ARMCortexR5x1CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexR5x1CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexR5x1CT.cpu0.l1icache	PVCache
ARMCortexR5x1CT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexR5x1CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexR5x1CT.ext_bus	PVBusLogger
ARMCortexR5x1CT.ext_bus.mapper	PVBusMapper
ARMCortexR5x1CT.l1_incoherent_interconnect	PVCache
ARMCortexR5x1CT.l1_incoherent_interconnect.downstream[0].pvbusmaster	PVBusMaster
ARMCortexR5x1CT.l1_incoherent_interconnect.upstream[U] (where U = 0-17)	PVBusSlave
ARMCortexR5x1CT.l2_flusher	AsyncCacheFlushUnit

## Ports for ARMCortexR5x1CT

Port	Direction	Protocol	Description
acp_s	slave	PVBus	ACP slave port.
cfgatcmsz	slave	Value	ATCM size.
cfgbtcmsz	slave	Value	BTCM Size.
cfgend	slave	Signal	This signal is for EE bit initialisation. This is CFGEE in RTL but cfgend here fastsim consistency reasons.
cfgnmfi	slave	Signal	Controls non-maskable Fast Interrupts.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
cpuhalt	slave	Signal	Raising this signal will put the core into halt mode.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.

Port	Direction	Protocol	Description
fiq	slave	Signal	This signal drives the CPU's fast-interrupt handling.
groupid	slave	Value	Group ID used for MPIDR.
initrama	slave	Signal	If ATCM is enabled at reset.
initramb	slave	Signal	If BTCM is enabled at reset.
irq	slave	Signal	This signal drives the CPU's interrupt handling.
loczrama	slave	Signal	Location of ATCM at reset.
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
pvbus_m	master	PVBus	The core will generate bus requests on this port.
pvbus_s	slave	PVBus	tcm slave port.
reset	slave	Signal	Raising this signal will put the core into reset mode.
standbywfe	master	Signal	This signal indicate if a core is in wfe state RTL calls this WFEPIPESTOPPED.
standbywfi	master	Signal	This signal indicates if a core is in WFI state RTL uses WFIPIPESTOPPED.
teinit	slave	Signal	Default exception handling state.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vic_ack	master	Signal	Vic acknowledge port to primary VIC.
vic_addr	slave	Value	Vic address port from primary VIC.
vinithi	slave	Signal	This signal controls of the location of the exception vectors at reset.

## Parameters for ARM Cortex R5x1CT

### **cpu0 . CFGATCMSZ**

Sets the size of the ATCM.

Type: `uint32_t`

Default value: `0xE`

### **cpu0 . CFGBTCMSZ**

Sets the size of the BTCM.

Type: `uint32_t`

Default value: `0xE`

### **cpu0 . CFGEND**

Initialize to BE8 endianness.

Type: `bool`

Default value: `false`

### **cpu0 . CFGIE**

Set the reset value of the instruction endian bit.



Type: `bool`

Default value: `false`

### **`cpu0.CFGNMFI`**

Enable nonmaskable FIQ interrupts on startup.

Type: `bool`

Default value: `false`

### **`cpu0.DP_FLOAT`**

Sets whether double-precision instructions are available.

Type: `bool`

Default value: `true`

### **`cpu0.LOCZRAMA`**

Initialize with LOCZRAMA set to 1.

Type: `bool`

Default value: `false`

### **`cpu0.TEINIT`**

T32 exception enable. The default has exceptions including reset handled in A32 state.

Type: `bool`

Default value: `false`

### **`cpu0.VINITHI`**

Initialize with high vectors enabled.

Type: `bool`

Default value: `false`

### **`cpu0.atcm_base`**

Model-specific. Sets the base address of the ATCM (forced to 0 if LOCZRAMA is 1).

Type: `uint32_t`

Default value: `0x00000000`

### **`cpu0.btcn_base`**

Model-specific. Sets the base address of the BTCN (forced to 0 if LOCZRAMA is 0).

Type: `uint32_t`

Default value: `0x00800000`

### **`cpu0.dcache-size`**

Set D-cache size in bytes.

Type: `uint32_t`

Default value: `0x10000`

### **`cpu0.icache-size`**

Set I-cache size in bytes.

Type: `uint32_t`

Default value: `0x10000`

### **`cpu0.min_sync_level`**

force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint32_t`

Default value: `0`

### **`cpu0.semihosting-ARM_HLT`**

ARM HLT number for semihosting.

Type: `uint32_t`

Default value: `0xF000`

### **`cpu0.semihosting-ARM_SVC`**

ARM SVC number for semihosting.

Type: `uint32_t`

Default value: `0x123456`

### **`cpu0.semihosting-Thumb_HLT`**

Thumb HLT number for semihosting.

Type: `uint32_t`

Default value: `0x3c`

### **`cpu0.semihosting-Thumb_SVC`**

Thumb SVC number for semihosting.

Type: `uint32_t`

Default value: `0xAB`

### **`cpu0.semihosting-cmd_line`**

Command line available to semihosting SVC calls.

Type: `string`

Default value: `""`

### **`cpu0.semihosting-cwd`**

Virtual address of CWD.

Type: `string`

Default value: `""`

### **`cpu0.semihosting-enable`**

Enable semihosting SVC Boolean true or false true/traps. Applications that do not use semihosting must set this parameter to False.

Type: `bool`

Default value: `true`

### **`cpu0.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint32_t`

Default value: `0x0`

### **`cpu0.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint32_t`

Default value: `0x0F000000`

### **`cpu0.semihosting-hlt-enable`**

Enable semihosting HLT traps. Applications that use HLT semihosting must set this parameter to true and the semihosting-enable parameter to true.

Type: `bool`

Default value: `false`

**cpu0.semihosting-prefix**

Prefix semihosting output with target instance name.

Type: `bool`

Default value: `false`

**cpu0.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint32_t`

Default value: `0x10000000`

**cpu0.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint32_t`

Default value: `0x0F000000`

**cpu0.vfp-enable\_at\_reset**

Enable coprocessor access and VFP at reset.

Type: `bool`

Default value: `false`

**cpu0.vfp-present**

Set whether model has VFP support.

Type: `bool`

Default value: `true`

**GROUP\_ID**

Value read in GROUP ID register field, bits[15:8] of the MPIDR.

Type: `uint32_t`

Default value: `0x0`

**INST\_ENDIAN**

Controls whether the model supports the instruction endianness bit.

Type: `bool`

Default value: `true`

**LOCK\_STEP**

Affects dual-processor configurations only, and ignored by single-processor configurations.

Type: `uint32_t`

Default value: `0`

**MICRO\_SCU**

Controls whether the effects of the MicroSCU are modeled.

Type: `bool`

Default value: `true`

**NUM\_BREAKPOINTS**

Controls with how many breakpoint pairs the model has been configured. This only affects the build options registers, because debug is not modeled.

Type: `uint32_t`

Default value: `0x7`

**NUM\_MPU\_REGION**

Sets the number of MPU regions.

Type: `uint32_t`

Default value: `12`

**NUM\_WATCHPOINTS**

Controls with how many watchpoint pairs the model has been configured. This only affects the build options registers, because debug is not modeled.

Type: `uint32_t`

Default value: `0x7`

**SLSPLIT**

Sets whether the model starts in split mode or locked mode.

Type: `bool`

Default value: `false`

**cpi\_div**

divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: 1

### **`cpi_mul`**

multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: 1

### **`dcache-state_modelled`**

Set whether D-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **`icache-state_modelled`**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **`reported_fp_revision`**

Purely cosmetic patch level value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored. Updates the FPSID.revision value.

Type: `int`

Default value: -1

### **`reported_patch_level`**

Purely cosmetic patch level value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: `int`

Default value: -1

### **`reported_revision_number`**

Purely cosmetic revision number value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: `int`

Default value: -1

## 3.94 ARMCortexR7x1CT

Defined in `LISA/ARMCortexR7x1CT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### About ARMCortexR7x1CT

An ARMCortexR7x2CT component also exists.

The per-core parameters are preceded by `cpun.`, where `n` identifies the core (0 or 1).

`pvbuss` is the slave port to access the TCM RAM of CPU `n`. Bits [3:0] of the user flags in the transaction are used to select the TCM:

**0**

Selects the ITCM of CPU 0

**1**

Selects the DTCM of CPU 0

**2**

Selects the ITCM of CPU 1

**3**

Selects the DTCM of CPU 1

Any other value is reserved.

When you use the `semihosting-cmd_line` parameter to set the command line that is available to semihosting SVC calls, the value of `argv[0]` is set to the first command-line argument, not to the name of an image.

### Differences between the model and the RTL

This component has the following differences from the corresponding revision of the RTL implementation:

- This component does not implement address filtering within the SCU. The enable bit for this feature is ignored.
- The GIC does not respect the `CFGSDISABLE` signal. This leads to some registers being accessible when they must not be.
- The SCU enable bit is ignored. The SCU is always enabled.

- The SCU ignores the invalidate all register.
- The Broadcast TLB or cache operations in this model do not cause other cores in the cluster that are asleep because of WFI to wake up.
- The RR bit in the SCTL is ignored.
- The Power Control Register in the system control coprocessor is implemented but writing to it does not change the behavior of the model.
- The model cannot be configured with a 128-entry TLB.
- When modeling the SCU, coherency operations are represented by a memory write followed by a read to refill from memory, rather than using cache-to-cache transfers.
- TCMs are modeled internally and the model does not support external TCMs or the ports associated with them.
- This component implements L1 cache as architecturally defined, but does not implement L2 cache. If you require an L2 cache you can add a PL310 Level 2 Cache Controller component.
- ECC and parity schemes are hardware-specific so are not supported.
- The `vfp-enable_at_reset` option is model-specific and has no hardware equivalent. Arm recommends that it is only used in test systems and tied off to false in production systems.

## Iris and MTI instances for ARM Cortex R7x1CT

This model has the following Iris instances:

Name	Instance type
ARMCortexR7x1CT	Cluster_ARM_Cortex-R7
ARMCortexR7x1CT.ARM_CortexR7x1CT.debug_rom	debug_rom
ARMCortexR7x1CT.acp_mapper	PVBusMapper
ARMCortexR7x1CT.cpu0	ARM_Cortex-R7
ARMCortexR7x1CT.cpu0.l1dcache	PVCache
ARMCortexR7x1CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexR7x1CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexR7x1CT.cpu0.l1icache	PVCache
ARMCortexR7x1CT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexR7x1CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexR7x1CT.ext_bus	PVBusLogger
ARMCortexR7x1CT.ext_bus.mapper	PVBusMapper
ARMCortexR7x1CT.l1_incoherent_interconnect	PVCache
ARMCortexR7x1CT.l1_incoherent_interconnect.downstream[0].pvbusmaster	PVBusMaster
ARMCortexR7x1CT.l1_incoherent_interconnect.upstream[U] (where U = 0-17)	PVBusSlave
ARMCortexR7x1CT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

Name	Component type
ARMCortexR7x1CT.acp_mapper	PVBusMapper



Name	Component type
ARMCortexR7x1CT.cpu0	ARM_Cortex-R7
ARMCortexR7x1CT.cpu0.l1dcache	PVCache
ARMCortexR7x1CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexR7x1CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexR7x1CT.cpu0.l1icache	PVCache
ARMCortexR7x1CT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexR7x1CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexR7x1CT.ext_bus	PVBusLogger
ARMCortexR7x1CT.ext_bus.mapper	PVBusMapper
ARMCortexR7x1CT.l1_incoherent_interconnect	PVCache
ARMCortexR7x1CT.l1_incoherent_interconnect.downstream[0].pvbusmaster	PVBusMaster
ARMCortexR7x1CT.l1_incoherent_interconnect.upstream[U] (where U = 0–17)	PVBusSlave
ARMCortexR7x1CT.l2_flusher	AsyncCacheFlushUnit

### Ports for ARMCortexR7x1CT

Port	Direction	Protocol	Description
acp_s	slave	PVBus	AXI ACP slave port.
cfgend	slave	Signal	This signal is for EE bit initialisation.
cfgnmfi	slave	Signal	This signal disables FIQ mask in CPSR.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
clusterid	slave	Value	The port sets the value in CPU ID register field, bits[11:8] of the MPIDR.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	Legacy FIQ request input line.
fiqout	master	Signal	Output of individual processor nFIQ from the interrupt controller.
fpuflags	master	ValueState	Floating-Point Unit output flags.
halt	slave	Signal	Raising this signal will put the core into halt mode. Equivalent to the hardware nCPUHALT[N:0] signal.
initram	slave	Signal	This signal enables the processor to boot from the instruction TCM.
ints	slave	Signal	Interrupt distributor interrupt lines.
irq	slave	Signal	Legacy IRQ request input line.
irqout	master	Signal	Output of individual processor nIRQ from the interrupt controller.
mfilteren	slave	Signal	This signal enables filtering of address ranges between master bus ports.
mfilterend	slave	Value	This port sets end of region mapped to pvbus_m1.
mfilterstart	slave	Value	This port sets start of region mapped to pvbus_m1.
periphbase	slave	Value	This port sets the base address of private peripheral region.
periphclk_in	slave	ClockSignal	The timer and the watchdog take need a clk that is scaled down at least by factor of two.
periphreset	slave	Signal	This signal resets timer and interrupt controller.
pfilterend	slave	Value	This port sets end of region mapped to pvbus_mp.
pfilterstart	slave	Value	This port sets start of region mapped to pvbus_mp.
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.

Port	Direction	Protocol	Description
pvbus_m0	master	PVBus	The core will generate bus requests on this port.
pvbus_m1	master	PVBus	The core will generate bus requests on this port.
pvbus_mp	master	PVBus	The core will generate bus requests on this port.
pvbus_s	slave	PVBus	tcm slave port
reset	slave	Signal	Raising this signal will put the core into reset mode.
scureset	slave	Signal	This signal resets SCU.
smpnamp	master	Signal	This signals AMP or SMP mode for each Cortex-R7 processor.
standbywfe	master	Signal	This signal indicates if a core is in WFE state.
standbywfi	master	Signal	This signal indicates if a core is in WFI state.
teinit	slave	Signal	This signal provides default exception handling state.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vinithi	slave	Signal	This signal controls of the location of the exception vectors at reset.
wdreset	slave	Signal	This signal resets individual watchdog.
wdresetreq	master	Signal	CPU watchdog reset requests.

## Parameters for ARMCortexR7x1CT

### **cpu0 . CFGEND**

Initialize to BE8 endianness.

Type: `bool`

Default value: `false`

### **cpu0 . CFGNMFI**

Enable nonmaskable FIQ interrupts on startup.

Type: `bool`

Default value: `false`

### **cpu0 . DP\_FLOAT**

Sets whether double-precision instructions are available.

Type: `bool`

Default value: `true`

### **cpu0 . INITRAM**

Enable the processor to boot from the instruction TCM.

Type: `bool`

Default value: `false`

**cpu0 . POWERCTLI**

Default power control state for processor.

Type: `uint32_t`

Default value: 0

**cpu0 . SMPnAMP**

Set whether the processor is part of a coherent domain.

Type: `bool`

Default value: `false`

**cpu0 . TEINIT**

T32 exception enable. The default has exceptions including reset handled in A32 state.

Type: `bool`

Default value: `false`

**cpu0 . VINITHI**

Initialize with high vectors enabled.

Type: `bool`

Default value: `false`

**cpu0 . dcache-size**

Set D-cache size in bytes.

Type: `uint32_t`

Default value: `0x8000`

**cpu0 . dtcm\_size**

Size of DTCM in KB.

Type: `uint32_t`

Default value: `0x8`

**cpu0 . icache-size**

Set I-cache size in bytes.

Type: `uint32_t`

Default value: `0x8000`

**cpu0.itcm\_size**

Size of ITCM in KB.

Type: uint32\_t

Default value: 0x8

**cpu0.min\_sync\_level**

force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint32\_t

Default value: 0

**cpu0.semihosting-ARM\_HLT**

ARM HLT number for semihosting.

Type: uint32\_t

Default value: 0xF000

**cpu0.semihosting-ARM\_SVC**

ARM SVC number for semihosting.

Type: uint32\_t

Default value: 0x123456

**cpu0.semihosting-Thumb\_HLT**

Thumb HLT number for semihosting.

Type: uint32\_t

Default value: 0x3c

**cpu0.semihosting-Thumb\_SVC**

Thumb SVC number for semihosting.

Type: uint32\_t

Default value: 0xAB

**cpu0.semihosting-cmd\_line**

Command line available to semihosting SVC calls.

Type: string

Default value: ""

**cpu0.semihosting-cwd**

Virtual address of CWD.

Type: `string`

Default value: `""`

**cpu0.semihosting-enable**

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to `false`.

Type: `bool`

Default value: `true`

**cpu0.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint32_t`

Default value: `0x0`

**cpu0.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint32_t`

Default value: `0x0F000000`

**cpu0.semihosting-hlt-enable**

Enable semihosting HLT traps. Applications that use HLT semihosting must set this parameter to `true` and the `semihosting-enable` parameter to `true`.

Type: `bool`

Default value: `false`

**cpu0.semihosting-prefix**

Prefix semihosting output with target instance name.

Type: `bool`

Default value: `false`

**cpu0.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint32_t`

Default value: `0x10000000`

### **`cpu0.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint32_t`

Default value: `0x0F000000`

### **`cpu0.tcm-present`**

Disables the DTCM and ITCM.

Type: `bool`

Default value: `true`

### **`cpu0.vfp-enable_at_reset`**

Enable coprocessor access and VFP at reset.

Type: `bool`

Default value: `false`

### **`cpu0.vfp-present`**

Set whether model has VFP support.

Type: `bool`

Default value: `true`

### **`CLUSTER_ID`**

Processor cluster ID value.

Type: `uint32_t`

Default value: `0x0`

### **`LOCK_STEP`**

Affects dual-processor configurations only, and ignored by single-processor configurations. 0 - Disable. Set for two independent processors. 1 - Lock Step. Appears to the system as two processors but is internally modeled as a single processor. 3 - Split Lock. Appears to the system as two processors but can be statically configured from reset either as two independent processors or two locked processors. For the model, these are equivalent to Disable and Lock Step, respectively, except for the value of build options registers. The model does not support dynamically splitting and locking the processor.

Type: `uint32_t`

Default value: 0

#### **MFILTEREN**

Enables filtering of address ranges.

Type: `bool`

Default value: `false`

#### **MFILTEREND**

Specifies the end address for address filtering.

Type: `uint32_t`

Default value: `0x0`

#### **MFILTERSTART**

Specifies the start address for address filtering.

Type: `uint32_t`

Default value: `0x0`

#### **NUM\_MPU\_REGION**

Sets the number of MPU regions.

Type: `uint32_t`

Default value: 12

#### **PERIPHBASE**

Base address of peripheral memory space.

Type: `uint32_t`

Default value: `0xAE000000`

#### **PFILTEREND**

Specifies the end address for peripheral port address filtering.

Type: `uint32_t`

Default value: `0x0`

#### **PFILTERSTART**

Specifies the start address for peripheral port address filtering.

Type: `uint32_t`

Default value: `0xFFFF0000`

### **`cpi_div`**

divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `1`

### **`cpi_mul`**

multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `1`

### **`dcache-state_modelled`**

Set whether D-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **`dic-spi_count`**

Number of shared peripheral interrupts implemented.

Type: `uint32_t`

Default value: `0x40`

### **`ecc_on`**

Enable Error Correcting Code.

Type: `bool`

Default value: `false`

### **`icache-state_modelled`**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`



**reported\_fp\_revision**

Purely cosmetic patch level value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored. Updates the FPSID.revision value.

Type: `int`

Default value: -1

**reported\_patch\_level**

Purely cosmetic patch level value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: `int`

Default value: -1

**reported\_revision\_number**

Purely cosmetic revision number value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: `int`

Default value: -1

## 3.95 ARMCortexR82AECT

Defined in `LISA/ARMCortexR82AECT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following parameters were added:

- `has_delayed_dbgreg`
- `has_delayed_sysreg`

### AE-specific features implemented

The only AE-specific features implemented in the model are Split-Lock and Lock-Step, with the following limitations:

- The implementation of Split-Lock and Lock-Step in the model is simply a halving of the number of cores available.
- Lock-step is implemented by a parameter that can be true or false for the entire execution only, rather than by a signal that can be enabled during execution.

As the model does not implement true Split-Lock functionality, the ports and parameters are contiguous. For example, for a 4-core CPU with Split-Lock enabled, `cpu0` and `cpu1` identify the available cores and associated ports, not `cpu0` and `cpu2`.

- Hybrid mode is not modeled in the DSU.

## Limitations

The model supports read and write access to the registers `IMP_MEMPROTCTLR_EL1` and `IMP_CLUSTERMEMPROTCTLR_EL1`, with no functional behavior implemented.

## Iris and MTI instances for ARM CortexR82AECT

This model has the following Iris instances:

Name	Instance type
ARMCortexR82AECT	Cluster_ARM_Cortex-R82AE
ARMCortexR82AECT.AMU	PVBusLogger
ARMCortexR82AECT.AMU.mapper	PVBusMapper
ARMCortexR82AECT.DAP	PVBusLogger
ARMCortexR82AECT.DAP.mapper	PVBusMapper
ARMCortexR82AECT.DSU	SharedR
ARMCortexR82AECT.DSU.PPU_cluster	PPUv1
ARMCortexR82AECT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexR82AECT.DSU.PPU_core0	PPUv1
ARMCortexR82AECT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexR82AECT.DSU.utility_slave[0]	PVBusSlave
ARMCortexR82AECT.MMAP	PVBusLogger
ARMCortexR82AECT.MMAP.mapper	PVBusMapper
ARMCortexR82AECT.RAS	PVBusLogger
ARMCortexR82AECT.RAS.mapper	PVBusMapper
ARMCortexR82AECT.cpu0	ARM_Cortex-R82AE
ARMCortexR82AECT.cpu0.UTLB	TLB
ARMCortexR82AECT.cpu0.debug_rom	debug_rom
ARMCortexR82AECT.cpu0.dtlb	TLB
ARMCortexR82AECT.cpu0.gicv3_cpu_if	GICv3CPUInterface
ARMCortexR82AECT.cpu0.l1dcache	PVCache
ARMCortexR82AECT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexR82AECT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexR82AECT.cpu0.l1licache	PVCache
ARMCortexR82AECT.cpu0.l1licache.downstream[0].pvbusmaster	PVBusMaster

Name	Instance type
ARMCortexR82AECT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexR82AECT.ext_bus	PVBusLogger
ARMCortexR82AECT.ext_bus.mapper	PVBusMapper
ARMCortexR82AECT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexR82AECT.global_debug_rom	debug_rom
ARMCortexR82AECT.l2_cache	PVCache
ARMCortexR82AECT.l2_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexR82AECT.l2_cache.upstream[Z] (where Z = 0-16)	PVBusSlave
ARMCortexR82AECT.llram_coherent_interconnect	PVCache
ARMCortexR82AECT.llram_coherent_interconnect.downstream[0].pvbusmaster	PVBusMaster
ARMCortexR82AECT.llram_coherent_interconnect.upstream[Y] (where Y = 0-2)	PVBusSlave
ARMCortexR82AECT.sbist_controllerChunk0	SBISTC0
ARMCortexR82AECT.secondary_debug_rom	debug_rom

This model has the following MTI trace components:

Name	Component type
ARMCortexR82AECT.AMU	PVBusLogger
ARMCortexR82AECT.AMU.mapper	PVBusMapper
ARMCortexR82AECT.DAP	PVBusLogger
ARMCortexR82AECT.DAP.mapper	PVBusMapper
ARMCortexR82AECT.DSU	SharedR
ARMCortexR82AECT.DSU.PPU_cluster	PPUv1
ARMCortexR82AECT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexR82AECT.DSU.PPU_core0	PPUv1
ARMCortexR82AECT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexR82AECT.DSU.utility_slave[0]	PVBusSlave
ARMCortexR82AECT.MMAP	PVBusLogger
ARMCortexR82AECT.MMAP.mapper	PVBusMapper
ARMCortexR82AECT.RAS	PVBusLogger
ARMCortexR82AECT.RAS.mapper	PVBusMapper
ARMCortexR82AECT.cpu0	ARM_Cortex-R82AE
ARMCortexR82AECT.cpu0.UTLB	TLB
ARMCortexR82AECT.cpu0.gicv3_cpu_if	GICv3CPUInterface
ARMCortexR82AECT.cpu0.l1dcache	PVCache
ARMCortexR82AECT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexR82AECT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexR82AECT.cpu0.l1licache	PVCache
ARMCortexR82AECT.cpu0.l1licache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexR82AECT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexR82AECT.ext_bus	PVBusLogger

Name	Component type
ARMCortexR82AECT.ext_bus.mapper	PVBusMapper
ARMCortexR82AECT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexR82AECT.l2_cache	PVCache
ARMCortexR82AECT.l2_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexR82AECT.l2_cache.upstream[Z] (where Z = 0-16)	PVBusSlave
ARMCortexR82AECT.llram_coherent_interconnect	PVCache
ARMCortexR82AECT.llram_coherent_interconnect.downstream[0].pvbusmaster	PVBusMaster
ARMCortexR82AECT.llram_coherent_interconnect.upstream[Y] (where Y = 0-2)	PVBusSlave

## Ports for ARMCortexR82AECT

Port	Direction	Protocol	Description
acel_s	slave	PVBus	External Slave port. Equivalent to AXIS port.
broadcastatomic	slave	Signal	CHI defined pins.
broadcastcachemaint	slave	Signal	Enable broadcasting of cache maintenance operations to downstream caches.
broadcastouter	slave	Signal	Enable broadcasting of Outer Shareable transactions.
cache_validation_control	slave	Value	This signal provides default exception handling state.
cfgendianess	slave	Signal	This signal if for EE bit initialisation
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
cluster_pcsn_pchannel	master	PChannel	Cluster PCSM signal
cluster_ppu_hw_stat	master	Value	Cluster PPU power state
clusterid	slave	Value	The port sets the value of the affinity levels 2 and 3; bits [39:32] and [23:16] of the MPIDR.
clusterpmuirq	master	Signal	DynamiQ pmu irq
CNTHPSIRQ	master	Signal	Timer signals to SOC
CNTHVSIRQ	master	Signal	Timer signals to SOC
CNTPSIRQ	master	Signal	Timer signals to SOC
cntvalueb	slave	CounterInterface	Interface to SoC level counter module
CNTVIRQ	master	Signal	Timer signals to SOC
commirq	master	Signal	Interrupt signal from debug communication channel.
complexcritirq	master	Signal	Complex RAS critical interrupt
complexerrirq	master	Signal	Complex RAS error interrupt
complexfaultirq	master	Signal	Complex RAS fault interrupt
core_pcsn_pchannel	master	PChannel	Core PCSM signals
core_ppu_hw_stat	master	Value	Notify the power state change inside the PPU.
coreerrirq	master	Signal	Core RAS error interrupt
corefaultirq	master	Signal	Core RAS fault interrupt
coreinstrrun	master	Signal	Core Running State

Port	Direction	Protocol	Description
cp15sdisable	slave	Signal	This signal disables write access to some system control processor registers
cpuhalt	slave	Signal	Raising this signal will put the core into halt mode.
cti0extin	slave	Signal	CTI trace inputs for core 0.
cti0extout	master	Signal	CTI trace outputs for core 0.
cti1extin	slave	Signal	CTI trace inputs for core 1.
cti1extout	master	Signal	CTI trace outputs for core 1.
cti2extin	slave	Signal	CTI trace inputs for core 2.
cti2extout	master	Signal	CTI trace outputs for core 2.
cti3extin	slave	Signal	CTI trace inputs for core 3.
cti3extout	master	Signal	CTI trace outputs for core 3.
cti4extin	slave	Signal	CTI trace inputs for core 4.
cti4extout	master	Signal	CTI trace outputs for core 4.
cti5extin	slave	Signal	CTI trace inputs for core 5.
cti5extout	master	Signal	CTI trace outputs for core 5.
cti6extin	slave	Signal	CTI trace inputs for core 6.
cti6extout	master	Signal	CTI trace outputs for core 6.
cti7extin	slave	Signal	CTI trace inputs for core 7.
cti7extout	master	Signal	CTI trace outputs for core 7.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	-
ctidbgirq	master	Signal	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	slave	Signal	-
dbgnopwrdown	master	Signal	These signals relate to core power down.
dbgpwrdownack	master	Signal	Debug power down acknowledge.
dbgpwrdownreq	slave	Signal	Debug power down request.
dev_debug_s	slave	PVBus	External debug interface.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
external_trace_reset	slave	Signal	ETMv4 External Trace Reset signal.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
l2reset	slave	Signal	This signal resets timer and interrupt controller and l2cache
llpp_m	master	PVBus	LLPP (Low-Latency Peripheral Port).
llram_m	master	PVBus	LLRAM Port
macp_s	slave	PVBus	MACP slave interface
memorymapped_debug_s	slave	PVBus	External debug interface.
periphbase	slave	Value_64	This port sets the base address of private peripheral region
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.

Port	Direction	Protocol	Description
ppu_cluster_irq	master	Signal	PPU cluster irq signal
ppu_cluster_isolate	master	Signal	PPU Cluster Domain isolation control.
ppu_cluster_wakerequest	slave	Signal	PPU cluster wakeup request
ppu_core_irq	master	Signal	PPU core irq signal
ppu_core_isolate	master	Signal	PPU Core Domain isolation control.
ppu_core_wakerequest	slave	Signal	PPU core wakeup request
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	master	PVBus	The core will generate bus requests on this port.
reset	slave	Signal	-
romaddr	slave	Value_64	Debug ROM base address.
romaddrv	slave	Signal	Debug ROM base address valid.
rvbar	slave	Value_64	Reset vector base address.
sei	slave	Signal	Per core System Error physical pins
spiden	slave	Signal	Secure invasive debug enable.
spp_m	master	PVBus	SPP (Shared Peripheral Port).
standbywfe	master	Signal	This signal indicates if a core is in WFE state
standbywfi	master	Signal	This signal indicates if a core is in WFI state
stlfailed	master	Signal	Indicates the STL has failed, set in any condition in which FFMIR.FMID has a failure mode set.
stlrunning	master	Signal	Indicates the STL is running, set in any condition in which FCTLR.STATUS is non-idle.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trace_unit_reset	slave	Signal	ETMv4 Trace Unit Reset signal.
utility_bus_s	slave	PVBus	Utility bus slave
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfiq	slave	Signal	This signal drives the CPUs virtual fast-interrupt handling.
virq	slave	Signal	This signal drives the CPUs virtual interrupt handling.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).

## Parameters for ARM Cortex R82AECT

### cpuX.CFGEND

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: `false`

**cpuX.CP15SDISABLE**

Initialize to disable access to some CP15 registers.

Type: `bool`

Default value: `false`

**cpuX.RVBAR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**cpuX.TEINIT**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**cpuX.ase-present**

Set whether the model has been built with NEON support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**cpuX.dcache-size**

L1 D-Cache size in bytes.

Type: `uint32_t`

Default value: `0x10000`

**cpuX.dtcn\_base**

Sets the 16K aligned base address of DTCM.

Type: `uint64_t`

Default value: `0x0`

**cpuX.dtcn\_size**

Sets the size of DTCM (in bytes).

Type: `uint32_t`

Default value: `0x0`

### **`cpuX.dtcn_stretch_clk`**

Whether DTCM clock stretched to occupy full cycle.

Type: `bool`

Default value: `false`

### **`cpuX.dtcn_wait`**

DTCM accesses wait states: 0-3 cycles.

Type: `uint32_t`

Default value: `0x0`

### **`cpuX.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`cpuX.icache-size`**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: `0x10000`

### **`cpuX.itcm_base`**

Sets the 16K aligned base address of ITCM.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.itcm_size`**

Sets the size of ITCM (in bytes).

Type: `uint32_t`

Default value: `0x0`

### **`cpuX.itcm_stretch_clk`**

Whether ITCM clock stretched to occupy full cycle.



Type: `bool`

Default value: `false`

### **`cpuX.itcm_wait`**

ITCM accesses wait states: 0-3 cycles.

Type: `uint32_t`

Default value: `0x0`

### **`cpuX.llpp.base`**

Equivalent to `CFGLLPPBASEADDR`.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.llpp.size`**

Equivalent to `CFGLLPPSIZE`.

Type: `uint32_t`

Default value: `0x8000000`

### **`cpuX.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

### **`cpuX.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

### **`cpuX.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**cpuX.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**cpuX.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**cpuX.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

**cpuX.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

**cpuX.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: `N/A`

**cpuX.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**cpuX.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**cpuX.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**cpuX.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**cpuX.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**cpuX.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**cpuX.tcm.a.enable**

Equivalent to `CFGITCMENm`.

Type: `bool`

Default value: `false`

**cpuX.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**cpuX.vfp-dp-present**

Whether double-precision floating point feature is implemented (FEAT\_F64MM).

Type: `bool`

Default value: `true`

### **`cpuX.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **`cpuX.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **`BROADCASTATOMIC`**

Enable broadcasting of atomic operation to NC/DEV memory. The broadcastatomic signal will override this value if used.

Type: `bool`

Default value: `true`

### **`BROADCASTATOMICL`**

Enable broadcasting of atomic operation to LLRAM memory. The broadcastatomicl signal will override this value and it will be functional for revision 2.

Type: `bool`

Default value: `true`

### **`BROADCASTCACHEMAINT`**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

Type: `bool`

Default value: `true`

### **`BROADCASTOUTER`**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

Type: `bool`

Default value: `true`

### **CCSIDR-L1D\_override**

If nonzero, override the value presented in CCSIDR for L1D (this is cosmetic and does not affect cache behaviour).

Type: `uint64_t`

Default value: `0x0`

### **CCSIDR-L1I\_override**

If nonzero, override the value presented in CCSIDR for L1I (this is cosmetic and does not affect cache behaviour).

Type: `uint64_t`

Default value: `0x0`

### **CCSIDR-L2\_override**

If nonzero, override the value presented in CCSIDR for L2 (this is cosmetic and does not affect cache behaviour).

Type: `uint64_t`

Default value: `0x0`

### **CCSIDR-L3\_override**

If nonzero, allow L3 selection in CSSELR and present this value in CCSIDR (this is cosmetic and does not affect cache behaviour).

Type: `uint64_t`

Default value: `0x0`

### **CFGTFPEN\_pin\_reset**

CFGTFPEN pin at reset.

Type: `bool`

Default value: `false`

### **CHI**

Selects the type of protocol the Main Manager(MM) interface implements. 0, MM port configured as AXI. 1, MM port configured as CHI.

Type: `bool`

Default value: `false`

**CLUSTER\_ID**

CLUSTER\_ID[15:8] equivalent to CFGMPIDRAFF3, CLUSTER\_ID[7:0] equivalent to CFGMPIDRAFF2.

Type: `uint16_t`

Default value: `0x0`

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. This value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain events even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required.

Type: `uint8_t`

Default value: `1`

**DBGROMADDR**

Initialization value of DBGDRAR register. Bits[55:12] of this register specify the ROM table physical address.

Type: `uint64_t`

Default value: `0x0`

**DBGROMADDRV**

If true, set bits[1:0] of the CP15 DBGDRAR to indicate that the address is valid.

Type: `bool`

Default value: `false`

**GICDISABLE**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`

Default value: `false`

**NUM\_CORES**

Number of cores in cluster.

Type: `uint8_t`

Default value: 1

**PA\_SIZE**

Physical address range supported (FEAT\_LPA).

Type: `uint8_t`

Default value: 40

**PERIPHBASE**

Base address of peripheral memory space.

Type: `uint64_t`

Default value: 0x13080000

**VMSA\_supported**

VMSA is supported at EL1.

Type: `bool`

Default value: `true`

**bus\_protection\_enable\_at\_reset**

Equivalent to CFGBUSPROTEN.

Type: `bool`

Default value: `false`

**core\_power\_on\_by\_default**

0 = Cluster PPU and all core PPUs reset to OFF, 1 = Cluster PPU and all core PPUs reset to ON.

Type: `bool`

Default value: `true`

**cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: 0x1

**cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: 0x1

### **dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **dcache-maintenance\_latency**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **dcache-miss\_latency**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **dcache-prefetch\_enabled**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

Type: bool

Default value: false

### **dcache-read\_access\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to



the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`dcache-snoop_data_transfer_latency`**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`dcache-state_modelled`**

Set whether D-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **`dcache-write_access_latency`**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`dcache-write_latency`**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`enable_lock_step`**

Whether the core is configured in Dual Core Lock Step mode (FEAT\_DCLS).

Type: `bool`

Default value: `false`

**enable\_simulation\_performance\_optimizations**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

Type: `bool`

Default value: `true`

**gicv3.BPR-min**

The minimum value for the GICC\_BPR register (non-secure version will be 1 + this value).

Type: `uint8_t`

Default value: 2

**gicv3.VBPR-min**

The minimum value for the GICV\_BPR register (non-secure version will be 1 + this value).

Type: `uint8_t`

Default value: 2

**has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

**has\_dense\_mem\_map**

If true, the cluster follows the dense memory map else it implements the sparse memory map.

Type: `bool`

Default value: `false`

**has\_impdef\_transient\_fault\_protection**

Support the Transient Fault Protection (TFP) flop parity errors through RAS registers (FEAT\_TFP).

Type: `bool`

Default value: `true`

### **has\_llpp**

Equivalent to CFGLLPPIMP.

Type: `bool`

Default value: `true`

### **has\_pmc**

Programmable MBIST controllers implemented.

Type: `bool`

Default value: `false`

### **has\_spp**

Equivalent to CFGSPPIMP.

Type: `bool`

Default value: `true`

### **icache-hit\_latency**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

Type: `bool`

Default value: `false`

**icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when icache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**icache-state\_modelled**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

**l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

### **`l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x400000`

### **`l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**memory.ext\_slave\_base**

Equivalent to `CFGACELSTCMBASEADDR`.

Type: `uint64_t`

Default value: `0x0`

**memory.has\_llram**

Equivalent to `CFGLLRAMIMP`.

Type: `bool`

Default value: `true`

**memory.llram\_base**

Equivalent to `CFGLLRAMBASEADDR`.

Type: `uint64_t`

Default value: `0x20000000`

**memory.llram\_enable\_at\_reset**

Equivalent to CFGLLRAMEN.

Type: `bool`

Default value: `true`

**memory.llram\_shared**

Equivalent to CFGLLRAMSHARED and it is only functional for revision 2.

Type: `bool`

Default value: `false`

**memory.llram\_size**

Size of the LLRAM.

Type: `uint32_t`

Default value: `0x10000000`

**num\_protection\_regions\_s1**

Number of v8-R protection regions.

Type: `uint8_t`

Default value: `16`

**num\_protection\_regions\_s2**

Number of v8-R hyp protection regions.

Type: `uint8_t`

Default value: `16`

**ptw\_latency**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

**ram\_protection\_enable\_at\_reset**

Equivalent to CFGRAMPROTEN.

Type: `bool`

Default value: `false`

**reported\_patch\_level**

Purely cosmetic patch level value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: `int8_t`

Default value: -1

**reported\_revision\_number**

Purely cosmetic revision number value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: `int8_t`

Default value: -1

**sbist\_controller.Chunk0.DL\_CYCLES**

DL\_CYCLES is the value that is loaded into FCTLR.FREQ out of reset.

Type: `uint8_t`

Default value: 160

**sbist\_controller.Chunk0.DL\_RESET**

DL\_RESET is used to initialize FCTLR.STATUS to INIT out of reset and load a value from the signal DL\_CYCLES into FCTLR.FREQ. This provides a method to enable deadlock checking out of reset, as it may be possible that a deadlock prevents initialization and setup of the SBIST controller.

Type: `uint8_t`

Default value: 0

**sbist\_controller.Chunk1.DL\_CYCLES**

DL\_CYCLES is the value that is loaded into FCTLR.FREQ out of reset.

Type: `uint8_t`

Default value: 160

**sbist\_controller.Chunk1.DL\_RESET**

DL\_RESET is used to initialize FCTLR.STATUS to INIT out of reset and load a value from the signal DL\_CYCLES into FCTLR.FREQ. This provides a method to enable deadlock checking out of reset, as it may be possible that a deadlock prevents initialization and setup of the SBIST controller.

Type: `uint8_t`

Default value: 0



**sbist\_controller.Chunk2.DL\_CYCLES**

DL\_CYCLES is the value that is loaded into FCTLR.FREQ out of reset.

Type: uint8\_t

Default value: 160

**sbist\_controller.Chunk2.DL\_RESET**

DL\_RESET is used to initialize FCTLR.STATUS to INIT out of reset and load a value from the signal DL\_CYCLES into FCTLR.FREQ. This provides a method to enable deadlock checking out of reset, as it may be possible that a deadlock prevents initialization and setup of the SBIST controller.

Type: uint8\_t

Default value: 0

**sbist\_controller.Chunk3.DL\_CYCLES**

DL\_CYCLES is the value that is loaded into FCTLR.FREQ out of reset.

Type: uint8\_t

Default value: 160

**sbist\_controller.Chunk3.DL\_RESET**

DL\_RESET is used to initialize FCTLR.STATUS to INIT out of reset and load a value from the signal DL\_CYCLES into FCTLR.FREQ. This provides a method to enable deadlock checking out of reset, as it may be possible that a deadlock prevents initialization and setup of the SBIST controller.

Type: uint8\_t

Default value: 0

**sbist\_controller.Chunk4.DL\_CYCLES**

DL\_CYCLES is the value that is loaded into FCTLR.FREQ out of reset.

Type: uint8\_t

Default value: 160

**sbist\_controller.Chunk4.DL\_RESET**

DL\_RESET is used to initialize FCTLR.STATUS to INIT out of reset and load a value from the signal DL\_CYCLES into FCTLR.FREQ. This provides a method to enable deadlock checking out of reset, as it may be possible that a deadlock prevents initialization and setup of the SBIST controller.

Type: uint8\_t

Default value: 0

**sbist\_controller.Chunk5.DL\_CYCLES**

DL\_CYCLES is the value that is loaded into FCTLR.FREQ out of reset.

Type: uint8\_t

Default value: 160

**sbist\_controller.Chunk5.DL\_RESET**

DL\_RESET is used to initialize FCTLR.STATUS to INIT out of reset and load a value from the signal DL\_CYCLES into FCTLR.FREQ. This provides a method to enable deadlock checking out of reset, as it may be possible that a deadlock prevents initialization and setup of the SBIST controller.

Type: uint8\_t

Default value: 0

**sbist\_controller.Chunk6.DL\_CYCLES**

DL\_CYCLES is the value that is loaded into FCTLR.FREQ out of reset.

Type: uint8\_t

Default value: 160

**sbist\_controller.Chunk6.DL\_RESET**

DL\_RESET is used to initialize FCTLR.STATUS to INIT out of reset and load a value from the signal DL\_CYCLES into FCTLR.FREQ. This provides a method to enable deadlock checking out of reset, as it may be possible that a deadlock prevents initialization and setup of the SBIST controller.

Type: uint8\_t

Default value: 0

**sbist\_controller.Chunk7.DL\_CYCLES**

DL\_CYCLES is the value that is loaded into FCTLR.FREQ out of reset.

Type: uint8\_t

Default value: 160

**sbist\_controller.Chunk7.DL\_RESET**

DL\_RESET is used to initialize FCTLR.STATUS to INIT out of reset and load a value from the signal DL\_CYCLES into FCTLR.FREQ. This provides a method to enable deadlock checking out of reset, as it may be possible that a deadlock prevents initialization and setup of the SBIST controller.

Type: uint8\_t

Default value: 0

**sbist\_controller.SBIST\_PAGE\_SIZE**

Size of the memory assigned to one Core in the SBIST Controller memory map.

Type: uint32\_t

Default value: 0x10000

**spp.base**

Equivalent to CFGSPPBASEADDR.

Type: uint64\_t

Default value: 0x0

**spp.size**

Sets the size of SPP(in bytes).

Type: uint32\_t

Default value: 0x8000000

**stage12\_tlb\_size**

If VMSA is supported at stage1, number of stage1+2 tlb entries. If instruction\_tlb\_size !=0, this is treated as dtlb size.

Type: uint32\_t

Default value: 0x0

**tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

**walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

## 3.96 ARMCortexR82CT

Defined in `LISA/ARMCortexR82CT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support
r1p1	Full support
r2p1	Full support
r3p1	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following parameters were added:

- `has_delayed_dbgreg`
- `has_delayed_sysreg`

### About ARMCortexR82CT

To simulate the r1p0 model, use the following parameters:

- `revision_number=1`
- `VMSA_supported=1`

### Limitations

The model supports read and write access to the registers `IMP_MEMPROTCTLR_EL1` and `IMP_CLUSTERMEMPROTCTLR_EL1`, with no functional behavior implemented.

Dense memory map support has been added for the Utility bus only.

### Iris and MTI instances for ARMCortexR82CT

This model has the following Iris instances:

Name	Instance type
<code>ARMCortexR82CT</code>	<code>Cluster_ARM_Cortex-R82</code>
<code>ARMCortexR82CT.AMU</code>	<code>PVBusLogger</code>
<code>ARMCortexR82CT.AMU.mapper</code>	<code>PVBusMapper</code>
<code>ARMCortexR82CT.DAP</code>	<code>PVBusLogger</code>
<code>ARMCortexR82CT.DAP.mapper</code>	<code>PVBusMapper</code>
<code>ARMCortexR82CT.DSU</code>	<code>SharedR</code>
<code>ARMCortexR82CT.DSU.PPU_cluster</code>	<code>PPUv1</code>
<code>ARMCortexR82CT.DSU.PPU_cluster.busslave</code>	<code>PVBusSlave</code>
<code>ARMCortexR82CT.DSU.PPU_core0</code>	<code>PPUv1</code>

Name	Instance type
ARMCortexR82CT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexR82CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexR82CT.MMAP	PVBusLogger
ARMCortexR82CT.MMAP.mapper	PVBusMapper
ARMCortexR82CT.RAS	PVBusLogger
ARMCortexR82CT.RAS.mapper	PVBusMapper
ARMCortexR82CT.cpu0	ARM_Cortex-R82
ARMCortexR82CT.cpu0.UTLB	TLB
ARMCortexR82CT.cpu0.debug_rom	debug_rom
ARMCortexR82CT.cpu0.dtlb	TLB
ARMCortexR82CT.cpu0.gicv3_cpu_if	GICv3CPUInterface
ARMCortexR82CT.cpu0.l1dcache	PVCache
ARMCortexR82CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexR82CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexR82CT.cpu0.l1icache	PVCache
ARMCortexR82CT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexR82CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexR82CT.ext_bus	PVBusLogger
ARMCortexR82CT.ext_bus.mapper	PVBusMapper
ARMCortexR82CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexR82CT.global_debug_rom	debug_rom
ARMCortexR82CT.l2_cache	PVCache
ARMCortexR82CT.l2_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexR82CT.l2_cache.upstream[Z] (where Z = 0-16)	PVBusSlave
ARMCortexR82CT.l1ram_coherent_interconnect	PVCache
ARMCortexR82CT.l1ram_coherent_interconnect.downstream[0].pvbusmaster	PVBusMaster
ARMCortexR82CT.l1ram_coherent_interconnect.upstream[Y] (where Y = 0-2)	PVBusSlave
ARMCortexR82CT.secondary_debug_rom	debug_rom

This model has the following MTI trace components:

Name	Component type
ARMCortexR82CT.AMU	PVBusLogger
ARMCortexR82CT.AMU.mapper	PVBusMapper
ARMCortexR82CT.DAP	PVBusLogger
ARMCortexR82CT.DAP.mapper	PVBusMapper
ARMCortexR82CT.DSU	SharedR
ARMCortexR82CT.DSU.PPU_cluster	PPUv1
ARMCortexR82CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexR82CT.DSU.PPU_core0	PPUv1
ARMCortexR82CT.DSU.PPU_core0.busslave	PVBusSlave

Name	Component type
ARMCortexR82CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexR82CT.MMAP	PVBusLogger
ARMCortexR82CT.MMAP.mapper	PVBusMapper
ARMCortexR82CT.RAS	PVBusLogger
ARMCortexR82CT.RAS.mapper	PVBusMapper
ARMCortexR82CT.cpu0	ARM_Cortex-R82
ARMCortexR82CT.cpu0.UTLB	TLB
ARMCortexR82CT.cpu0.gicv3_cpu_if	GICv3CPUInterface
ARMCortexR82CT.cpu0.l1dcache	PVCache
ARMCortexR82CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexR82CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexR82CT.cpu0.l1licache	PVCache
ARMCortexR82CT.cpu0.l1licache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexR82CT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexR82CT.ext_bus	PVBusLogger
ARMCortexR82CT.ext_bus.mapper	PVBusMapper
ARMCortexR82CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexR82CT.l2_cache	PVCache
ARMCortexR82CT.l2_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexR82CT.l2_cache.upstream[Z] (where Z = 0–16)	PVBusSlave
ARMCortexR82CT.llram_coherent_interconnect	PVCache
ARMCortexR82CT.llram_coherent_interconnect.downstream[0].pvbusmaster	PVBusMaster
ARMCortexR82CT.llram_coherent_interconnect.upstream[Y] (where Y = 0–2)	PVBusSlave

## Ports for ARMCortexR82CT

Port	Direction	Protocol	Description
acel_s	slave	PVBus	External Slave port. Equivalent to AXIS port.
broadcastatomic	slave	Signal	CHI defined pins.
broadcastatomicl	slave	Signal	BROADCASTATOMIC pin for LLRAM
broadcastcachemaint	slave	Signal	Enable broadcasting of cache maintenance operations to downstream caches.
broadcastouter	slave	Signal	Enable broadcasting of Outer Shareable transactions.
cache_validation_control	slave	Value	This signal provides default exception handling state.
cfgendianess	slave	Signal	This signal if for EE bit initialisation
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
cluster_pcsn_pchannel	master	PChannel	Cluster PCSM signal
cluster_ppu_hw_stat	master	Value	Cluster PPU power state

Port	Direction	Protocol	Description
clusterid	slave	Value	The port sets the value of the affinity levels 2 and 3; bits [39:32] and [23:16] of the MPIDR.
clusterpmuirq	master	Signal	DynamiQ pmu irq
CNTHPSIRQ	master	Signal	Timer signals to SOC
CNTHVSIRQ	master	Signal	Timer signals to SOC
CNTPSIRQ	master	Signal	Timer signals to SOC
cntvalueb	slave	CounterInterface	Interface to SoC level counter module
CNTVIRQ	master	Signal	Timer signals to SOC
commirq	master	Signal	Interrupt signal from debug communication channel.
core_pcsn_pchannel	master	PChannel	Core PCSM signals
core_ppu_hw_stat	master	Value	Notify the power state change inside the PPU.
coreerrirq	master	Signal	Core RAS error interrupt
corefaultirq	master	Signal	Core RAS fault interrupt
coreinstrrun	master	Signal	Core Running State
cp15sdisable	slave	Signal	This signal disables write access to some system control processor registers
cpuhalt	slave	Signal	Raising this signal will put the core into halt mode.
cti0extin	slave	Signal	CTI trace inputs for core 0.
cti0extout	master	Signal	CTI trace outputs for core 0.
cti1extin	slave	Signal	CTI trace inputs for core 1.
cti1extout	master	Signal	CTI trace outputs for core 1.
cti2extin	slave	Signal	CTI trace inputs for core 2.
cti2extout	master	Signal	CTI trace outputs for core 2.
cti3extin	slave	Signal	CTI trace inputs for core 3.
cti3extout	master	Signal	CTI trace outputs for core 3.
cti4extin	slave	Signal	CTI trace inputs for core 4.
cti4extout	master	Signal	CTI trace outputs for core 4.
cti5extin	slave	Signal	CTI trace inputs for core 5.
cti5extout	master	Signal	CTI trace outputs for core 5.
cti6extin	slave	Signal	CTI trace inputs for core 6.
cti6extout	master	Signal	CTI trace outputs for core 6.
cti7extin	slave	Signal	CTI trace inputs for core 7.
cti7extout	master	Signal	CTI trace outputs for core 7.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	-
ctidbgirq	master	Signal	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	slave	Signal	-
dbgnopwrdown	master	Signal	These signals relate to core power down.
dbgprwrdownack	master	Signal	Debug power down acknowledge.
dbgprwrdownreq	slave	Signal	Debug power down request.
dev_debug_s	slave	PVBus	External debug interface.

Port	Direction	Protocol	Description
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
external_trace_reset	slave	Signal	ETMv4 External Trace Reset signal.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
l2reset	slave	Signal	This signal resets timer and interrupt controller and l2cache
llpp_m	master	PVBus	LLPP (Low-Latency Peripheral Port).
llram_m	master	PVBus	LLRAM Port
macp_s	slave	PVBus	MACP slave interface
memorymapped_debug_s	slave	PVBus	External debug interface.
periphbase	slave	Value_64	This port sets the base address of private peripheral region
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	master	Signal	PPU cluster irq signal
ppu_cluster_isolate	master	Signal	PPU Cluster Domain isolation control.
ppu_cluster_wakerequest	slave	Signal	PPU cluster wakeup request
ppu_core_irq	master	Signal	PPU core irq signal
ppu_core_isolate	master	Signal	PPU Core Domain isolation control.
ppu_core_wakerequest	slave	Signal	PPU core wakeup request
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	master	PVBus	The core will generate bus requests on this port.
reset	slave	Signal	-
romaddr	slave	Value_64	Debug ROM base address.
romaddrv	slave	Signal	Debug ROM base address valid.
rvbar	slave	Value_64	Reset vector base address.
sei	slave	Signal	Per core System Error physical pins
spiden	slave	Signal	Secure invasive debug enable.
spp_m	master	PVBus	SPP (Shared Peripheral Port).
standbywfe	master	Signal	This signal indicates if a core is in WFE state
standbywfi	master	Signal	This signal indicates if a core is in WFI state
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trace_unit_reset	slave	Signal	ETMv4 Trace Unit Reset signal.
utility_bus_s	slave	PVBus	Utility bus slave
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfiq	slave	Signal	This signal drives the CPUs virtual fast-interrupt handling.



Port	Direction	Protocol	Description
virq	slave	Signal	This signal drives the CPUs virtual interrupt handling.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).

## Parameters for ARM CortexR82CT

### **cpuX.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **cpuX.CP15SDISABLE**

Initialize to disable access to some CP15 registers.

Type: `bool`

Default value: `false`

### **cpuX.RVBAR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **cpuX.TEINIT**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **cpuX.ase-present**

Set whether the model has been built with NEON support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

### **cpuX.dcache-size**

L1 D-Cache size in bytes.

Type: `uint32_t`

Default value: `0x10000`

### **`cpuX.dtcn_base`**

Sets the 16K aligned base address of DTCM.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.dtcn_size`**

Sets the size of DTCM (in bytes).

Type: `uint32_t`

Default value: `0x0`

### **`cpuX.dtcn_stretch_clk`**

Whether DTCM clock stretched to occupy full cycle.

Type: `bool`

Default value: `false`

### **`cpuX.dtcn_wait`**

DTCM accesses wait states: 0-3 cycles.

Type: `uint32_t`

Default value: `0x0`

### **`cpuX.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`cpuX.icache-size`**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: `0x10000`

### **`cpuX.itcm_base`**

Sets the 16K aligned base address of ITCM.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.itcm_size`**

Sets the size of ITCM (in bytes).

Type: `uint32_t`

Default value: `0x0`

### **`cpuX.itcm_stretch_clk`**

Whether ITCM clock stretched to occupy full cycle.

Type: `bool`

Default value: `false`

### **`cpuX.itcm_wait`**

ITCM accesses wait states: 0-3 cycles.

Type: `uint32_t`

Default value: `0x0`

### **`cpuX.llpp.base`**

Equivalent to `CFGLLPPBASEADDR`.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.llpp.size`**

Equivalent to `CFGLLPPSIZE`.

Type: `uint32_t`

Default value: `0x8000000`

### **`cpuX.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

**cpuX.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**cpuX.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**cpuX.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**cpuX.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**cpuX.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**cpuX.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**cpuX.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**cpuX.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

**cpuX.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**cpuX.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**cpuX.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**cpuX.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**cpuX.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**cpuX.tcm.a.enable**

Equivalent to CFGITCMENm.

Type: `bool`

Default value: `false`

**cpuX.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**cpuX.vfp-dp-present**

Whether double-precision floating point feature is implemented (FEAT\_F64MM).

Type: bool

Default value: true

**cpuX.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**cpuX.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: true

**BROADCASTATOMIC**

Enable broadcasting of atomic operation to NC/DEV memory. The broadcastatomic signal will override this value if used.

Type: bool

Default value: true

**BROADCASTATOMICL**

Enable broadcasting of atomic operation to LLRAM memory. The broadcastatomicl signal will override this value and it will be functional for revision 2.

Type: bool

Default value: true

**BROADCASTCACHEMAINT**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

Type: `bool`

Default value: `true`

**BROADCASTOUTER**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

Type: `bool`

Default value: `true`

**CCSIDR-L1D\_override**

If nonzero, override the value presented in CCSIDR for L1D (this is cosmetic and does not affect cache behaviour).

Type: `uint64_t`

Default value: `0x0`

**CCSIDR-L1I\_override**

If nonzero, override the value presented in CCSIDR for L1I (this is cosmetic and does not affect cache behaviour).

Type: `uint64_t`

Default value: `0x0`

**CCSIDR-L2\_override**

If nonzero, override the value presented in CCSIDR for L2 (this is cosmetic and does not affect cache behaviour).

Type: `uint64_t`

Default value: `0x0`

**CCSIDR-L3\_override**

If nonzero, allow L3 selection in CSSELR and present this value in CCSIDR (this is cosmetic and does not affect cache behaviour).

Type: `uint64_t`

Default value: `0x0`

**CHI**

Selects the type of protocol the Main Manager(MM) interface implements. 0, MM port configured as AXI. 1, MM port configured as CHI.

Type: `bool`

Default value: `false`

**CLUSTER\_ID**

CLUSTER\_ID[15:8] equivalent to CFGMPIDRAFF3, CLUSTER\_ID[7:0] equivalent to CFGMPIDRAFF2.

Type: `uint16_t`

Default value: `0x0`

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain even when cache state modelling is disabled. Possible values are:- 0 = All CMOs are broadcast if architecturally required- 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required.

Type: `uint8_t`

Default value: `1`

**DBGROMADDR**

Initialization value of DBGDRAR register. Bits[55:12] of this register specify the ROM table physical address.

Type: `uint64_t`

Default value: `0x0`

**DBGROMADDRV**

If true, set bits[1:0] of the CP15 DBGDRAR to indicate that the address is valid.

Type: `bool`

Default value: `false`

**GICDISABLE**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`



Default value: `false`

### **NUM\_CORES**

Number of cores in cluster.

Type: `uint8_t`

Default value: 1

### **PA\_SIZE**

Physical address range supported (FEAT\_LPA).

Type: `uint8_t`

Default value: 40

### **PERIPHBASE**

Base address of peripheral memory space.

Type: `uint64_t`

Default value: `0x13080000`

### **VMSA\_supported**

VMSA is supported at EL1.

Type: `bool`

Default value: `false`

### **core\_power\_on\_by\_default**

0 = Cluster PPU and all core PPUs reset to OFF, 1 = Cluster PPU and all core PPUs reset to ON.

Type: `bool`

Default value: `true`

### **cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

### **cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: 0x1

### **dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **dcache-maintenance\_latency**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **dcache-miss\_latency**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **dcache-prefetch\_enabled**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

Type: bool

Default value: false

### **dcache-read\_access\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to

the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`dcache-snoop_data_transfer_latency`**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`dcache-state_modelled`**

Set whether D-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **`dcache-write_access_latency`**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`dcache-write_latency`**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`enable_lock_step`**

Whether the core is configured in Dual Core Lock Step mode (FEAT\_DCLS).

Type: `bool`

Default value: `false`

**enable\_simulation\_performance\_optimizations**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

Type: `bool`

Default value: `true`

**gicv3.BPR-min**

The minimum value for the GICC\_BPR register (non-secure version will be 1 + this value).

Type: `uint8_t`

Default value: 2

**gicv3.VBPR-min**

The minimum value for the GICV\_BPR register (non-secure version will be 1 + this value).

Type: `uint8_t`

Default value: 2

**has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

**has\_dense\_mem\_map**

If true, the cluster follows the dense memory map else it implements the sparse memory map.

Type: `bool`

Default value: `false`

**has\_llpp**

Equivalent to CFGLLPPIMP.

Type: `bool`

Default value: `true`

### **has\_spp**

Equivalent to CFGSPPIIMP.

Type: `bool`

Default value: `true`

### **icache-hit\_latency**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

### **icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and

intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`icache-read_latency`**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`icache-state_modelled`**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **`l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x400000

**l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

### **`l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`memory.ext_slave_base`**

Equivalent to `CFGACELSTCMBASEADDR`.

Type: `uint64_t`

Default value: `0x0`

### **`memory.has_llram`**

Equivalent to `CFGLLRAMIMP`.

Type: `bool`

Default value: `true`

### **`memory.llram_base`**

Equivalent to `CFGLLRAMBASEADDR`.

Type: `uint64_t`

Default value: `0x20000000`

### **`memory.llram_enable_at_reset`**

Equivalent to `CFGLLRAMEN`.

Type: `bool`

Default value: `true`

### **`memory.llram_shared`**

Equivalent to `CFGLLRAMSHARED` and it is only functional for revision 2.

Type: `bool`

Default value: `false`



**memory.llram\_size**

Size of the LLRAM.

Type: `uint32_t`

Default value: `0x10000000`

**num\_protection\_regions\_s1**

Number of v8-R protection regions.

Type: `uint8_t`

Default value: `16`

**num\_protection\_regions\_s2**

Number of v8-R hyp protection regions.

Type: `uint8_t`

Default value: `16`

**patch\_level**

Patch level of TRM implemented by the model. Changing the value can change the model behaviour. Visible as the Revision field in MIDR/MIDR\_EL1. Corresponds to the patch number Y in rXpY.

Type: `uint8_t`

Default value: `1`

**ptw\_latency**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

**ram\_protection\_enable\_at\_reset**

Equivalent to CFGRAMPROTEN.

Type: `bool`

Default value: `false`

**revision\_number**

Revision number of TRM implemented by the model. Changing the value can change the model behaviour. Visible as the Variant field in MIDR/MIDR\_EL1. Corresponds to the revision number X in rXpY.

Type: uint8\_t

Default value: 1

**spp.base**

Equivalent to CFGSPPBSEADDR.

Type: uint64\_t

Default value: 0x0

**spp.size**

Sets the size of SPP(in bytes).

Type: uint32\_t

Default value: 0x8000000

**stage12\_tlb\_size**

If VMSA is supported at stage1, number of stage1+2 tlb entries. If instruction\_tlb\_size !=0, this is treated as dtlb size.

Type: uint32\_t

Default value: 0x0

**tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

**walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

## 3.97 ARMCortexR8x1CT

Defined in `LISA/ARMCortexR8x1CT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### About ARMCortexR8x1CT

The following components also exist:

ARMCortexR8x2CT. ARMCortexR8x3CT. ARMCortexR8x4CT.

The per-core parameters are preceded by `cpun.`, where `n` identifies the core (0-3).

`pvbuss` is the slave port to access the TCM RAM of CPU `n`. Bits [3:0] of the user flags in the transaction are used to select the TCM:

- 0**  
Selects the ITCM of CPU 0.
- 1**  
Selects the DTCM of CPU 0.
- 2**  
Selects the ITCM of CPU 1.
- 3**  
Selects the DTCM of CPU 1.
- 4**  
Selects the ITCM of CPU 2.
- 5**  
Selects the DTCM of CPU 2.
- 6**  
Selects the ITCM of CPU 3.
- 7**  
Selects the DTCM of CPU 3.

Any other value is reserved.

The `semihosting-cwd` parameter sets the CWD that is used for semihosting. The host operating system limits the maximum path length. The `semihosting-cwd` parameter does not provide any security. Software running on the model can access files outside this directory using relative paths containing `..` or using absolute paths.

## Differences between the model and the RTL

This component has the following differences from the corresponding revision of the RTL implementation:

- This component does not implement address filtering within the SCU. The enable bit for this feature is ignored.
- The GIC does not respect the `CFGSDISABLE` signal. This leads to some registers being accessible when they must not be.
- The SCU enable bit is ignored. The SCU is always enabled.
- The SCU ignores the invalidate all register.
- The Broadcast TLB or cache operations in this model do not cause other cores in the cluster that are asleep because of WFI to wake up.
- The RR bit in the SCTLR is ignored.
- The Power Control Register in the system control coprocessor is implemented but writing to it does not change the behavior of the model.
- The model cannot be configured with a 128-entry TLB.
- When modeling the SCU, coherency operations are represented by a memory write followed by a read to refill from memory, rather than using cache-to-cache transfers.
- TCMs are modeled internally and the model does not support external TCMs or the ports associated with them.
- This component implements L1 cache as architecturally defined, but does not implement L2 cache. If you require an L2 cache you can add a PL310 Level 2 Cache Controller component.
- The `vfp-enable_at_reset` option is model-specific and has no hardware equivalent. Arm recommends that it is only used in test systems and tied off to false in production systems.
- ECC and parity schemes are hardware-specific so are not supported.

## Iris and MTI instances for ARMCortexR8x1CT

This model has the following Iris instances:

Name	Instance type
ARMCortexR8x1CT	Cluster_ARM_Cortex-R8
ARMCortexR8x1CT.ARM_CortexR8x1CT.debug_rom	debug_rom
ARMCortexR8x1CT.acp_mapper	PVBusMapper
ARMCortexR8x1CT.cpu0	ARM_Cortex-R8
ARMCortexR8x1CT.cpu0.l1dcache	PVCache
ARMCortexR8x1CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexR8x1CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexR8x1CT.cpu0.l1icache	PVCache
ARMCortexR8x1CT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexR8x1CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexR8x1CT.ext_bus	PVBusLogger
ARMCortexR8x1CT.ext_bus.mapper	PVBusMapper

Name	Instance type
ARMCortexR8x1CT.l1_incoherent_interconnect	PVCache
ARMCortexR8x1CT.l1_incoherent_interconnect.downstream[0].pvbusmaster	PVBusMaster
ARMCortexR8x1CT.l1_incoherent_interconnect.upstream[U] (where U = 0-17)	PVBusSlave
ARMCortexR8x1CT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

Name	Component type
ARMCortexR8x1CT.acp_mapper	PVBusMapper
ARMCortexR8x1CT.cpu0	ARM_Cortex-R8
ARMCortexR8x1CT.cpu0.l1dcache	PVCache
ARMCortexR8x1CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexR8x1CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexR8x1CT.cpu0.l1icache	PVCache
ARMCortexR8x1CT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexR8x1CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexR8x1CT.ext_bus	PVBusLogger
ARMCortexR8x1CT.ext_bus.mapper	PVBusMapper
ARMCortexR8x1CT.l1_incoherent_interconnect	PVCache
ARMCortexR8x1CT.l1_incoherent_interconnect.downstream[0].pvbusmaster	PVBusMaster
ARMCortexR8x1CT.l1_incoherent_interconnect.upstream[U] (where U = 0-17)	PVBusSlave
ARMCortexR8x1CT.l2_flusher	AsyncCacheFlushUnit

## Ports for ARMCortexR8x1CT

Port	Direction	Protocol	Description
acp_s	slave	PVBus	AXI ACP slave port.
cfgend	slave	Signal	This signal is for EE bit initialisation.
cfgnmfi	slave	Signal	This signal disables FIQ mask in CPSR.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
clusterid	slave	Value	The port sets the value in CPU ID register field, bits[11:8] of the MPIDR.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	Legacy FIQ request input line.
fiqout	master	Signal	Output of individual processor nFIQ from the interrupt controller.
fpfilterend0	slave	Value	This port sets end of region mapped to pvbus_mfp0.
fpfilterend1	slave	Value	This port sets end of region mapped to pvbus_mfp1.
fpfilterend2	slave	Value	This port sets end of region mapped to pvbus_mfp2.
fpfilterend3	slave	Value	This port sets end of region mapped to pvbus_mfp3.
fpfilterstart0	slave	Value	This port sets start of region mapped to pvbus_mfp0.
fpfilterstart1	slave	Value	This port sets start of region mapped to pvbus_mfp1.
fpfilterstart2	slave	Value	This port sets start of region mapped to pvbus_mfp2.

Port	Direction	Protocol	Description
fpfilterstart3	slave	Value	This port sets start of region mapped to pvbus_mfp3.
fpuflags	master	ValueState	Floating-Point Unit output flags.
halt	slave	Signal	Raising this signal will put the core into halt mode. Equivalent to the hardware nCPUHALT[N:0] signal.
initram	slave	Signal	This signal enables the processor to boot from the instruction TCM.
ints	slave	Signal	Interrupt distributor interrupt lines.
irq	slave	Signal	Legacy IRQ request input line.
irqout	master	Signal	Output of individual processor nIRQ from the interrupt controller.
mfilteren	slave	Signal	This signal enables filtering of address ranges between master bus ports.
mfilterend	slave	Value	This port sets end of region mapped to pvbus_m1.
mfilterstart	slave	Value	This port sets start of region mapped to pvbus_m1.
periphbase	slave	Value	This port sets the base address of private peripheral region.
periphclk_in	slave	ClockSignal	The timer and the watchdog take need a clk that is scaled down at least by factor of two.
periphreset	slave	Signal	This signal resets timer and interrupt controller.
pfilterend	slave	Value	This port sets end of region mapped to pvbus_mp.
pfilterstart	slave	Value	This port sets start of region mapped to pvbus_mp.
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
pvbus_m0	master	PVBus	AXI master port 0.
pvbus_m1	master	PVBus	AXI master port 1.
pvbus_mfp0	master	PVBus	Fast peripheral port for core 0.
pvbus_mfp1	master	PVBus	Fast peripheral port for core 1.
pvbus_mfp2	master	PVBus	Fast peripheral port for core 2.
pvbus_mfp3	master	PVBus	Fast peripheral port for core 3.
pvbus_mp	master	PVBus	Shared peripheral port.
pvbus_s	slave	PVBus	tcm slave port.
reset	slave	Signal	Raising this signal will put the core into reset mode.
scureset	slave	Signal	This signal resets SCU.
smpnamp	master	Signal	This signals AMP or SMP mode for each Cortex-R8 processor.
standbywfe	master	Signal	This signal indicates if a core is in WFE state.
standbywfi	master	Signal	This signal indicates if a core is in WFI state.
teinit	slave	Signal	This signal provides default exception handling state.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vinithi	slave	Signal	This signal controls of the location of the exception vectors at reset.
wdreset	slave	Signal	This signal resets individual watchdog.
wdresetreq	master	Signal	CPU watchdog reset requests.

## Parameters for ARMCortexR8x1CT

### cpu0 . CFGEND

Initialize to BE8 endianness.

Type: `bool`

Default value: `false`

### **`cpu0.CFGNMFI`**

Enable nonmaskable FIQ interrupts on startup.

Type: `bool`

Default value: `false`

### **`cpu0.DP_FLOAT`**

Sets whether double-precision instructions are available.

Type: `bool`

Default value: `true`

### **`cpu0.INITRAM`**

Enable the processor to boot from the instruction TCM.

Type: `bool`

Default value: `false`

### **`cpu0.POWERCTLI`**

Default power control state for processor.

Type: `uint32_t`

Default value: 0

### **`cpu0.SMPnAMP`**

Set whether the processor is part of a coherent domain.

Type: `bool`

Default value: `false`

### **`cpu0.TEINIT`**

T32 exception enable. The default has exceptions including reset handled in A32 state.

Type: `bool`

Default value: `false`

### **`cpu0.VINITHI`**

Initialize with high vectors enabled.

Type: `bool`

Default value: `false`

### **`cpu0.dcache-size`**

Set D-cache size in bytes.

Type: `uint32_t`

Default value: `0x8000`

### **`cpu0.dtcn_size`**

Size of DTCM in KB.

Type: `uint32_t`

Default value: `0x8`

### **`cpu0.icache-size`**

Set I-cache size in bytes.

Type: `uint32_t`

Default value: `0x8000`

### **`cpu0.itcn_size`**

Size of ITCM in KB.

Type: `uint32_t`

Default value: `0x8`

### **`cpu0.min_sync_level`**

force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint32_t`

Default value: `0`

### **`cpu0.semihosting-ARM_HLT`**

ARM HLT number for semihosting.

Type: `uint32_t`

Default value: `0xF000`

### **`cpu0.semihosting-ARM_SVC`**

ARM SVC number for semihosting.



Type: `uint32_t`

Default value: `0x123456`

### **`cpu0.semihosting-Thumb_HLT`**

Thumb HLT number for semihosting.

Type: `uint32_t`

Default value: `0x3c`

### **`cpu0.semihosting-Thumb_SVC`**

Thumb SVC number for semihosting.

Type: `uint32_t`

Default value: `0xAB`

### **`cpu0.semihosting-cmd_line`**

Command line available to semihosting SVC calls.

Type: `string`

Default value: `""`

### **`cpu0.semihosting-cwd`**

Virtual address of CWD.

Type: `string`

Default value: `""`

### **`cpu0.semihosting-enable`**

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

Type: `bool`

Default value: `true`

### **`cpu0.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint32_t`

Default value: `0x0`

**cpu0.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint32_t`

Default value: `0x0F000000`

**cpu0.semihosting-hlt-enable**

Enable semihosting HLT traps. Applications that use HLT semihosting must set this parameter to true and the semihosting-enable parameter to true.

Type: `bool`

Default value: `false`

**cpu0.semihosting-prefix**

Prefix semihosting output with target instance name.

Type: `bool`

Default value: `false`

**cpu0.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint32_t`

Default value: `0x10000000`

**cpu0.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint32_t`

Default value: `0x0F000000`

**cpu0.tcm-present**

Disables the DTCM and ITCM.

Type: `bool`

Default value: `true`

**cpu0.vfp-enable\_at\_reset**

Enable coprocessor access and VFP at reset.

Type: `bool`

Default value: `false`

### **`cpu0.vfp-present`**

Set whether model has VFP support.

Type: `bool`

Default value: `true`

### **`CLUSTER_ID`**

Processor cluster ID value.

Type: `uint32_t`

Default value: `0x0`

### **`FPFILTEREND0`**

Specifies the end address for fast peripheral port address filtering.

Type: `uint32_t`

Default value: `0x0`

### **`FPFILTEREND1`**

Specifies the end address for fast peripheral port address filtering.

Type: `uint32_t`

Default value: `0x0`

### **`FPFILTEREND2`**

Specifies the end address for fast peripheral port address filtering.

Type: `uint32_t`

Default value: `0x0`

### **`FPFILTEREND3`**

Specifies the end address for fast peripheral port address filtering.

Type: `uint32_t`

Default value: `0x0`

### **`FPFILTERSTART0`**

Specifies the start address for fast peripheral port address filtering.

Type: `uint32_t`

Default value: 0xFFFF0000

**FPFILTERSTART1**

Specifies the start address for fast peripheral port address filtering.

Type: uint32\_t

Default value: 0xFFFF0000

**FPFILTERSTART2**

Specifies the start address for fast peripheral port address filtering.

Type: uint32\_t

Default value: 0xFFFF0000

**FPFILTERSTART3**

Specifies the start address for fast peripheral port address filtering.

Type: uint32\_t

Default value: 0xFFFF0000

**LOCK\_STEP**

Affects dual-processor configurations only, and ignored by single-processor configurations. 0 - Disable. Set for two independent processors. 1 - Lock Step. Appears to the system as two processors but is internally modeled as a single processor. 3 - Split Lock. Appears to the system as two processors but can be statically configured from reset either as two independent processors or two locked processors. For the model, these are equivalent to Disable and Lock Step, respectively, except for the value of build options registers. The model does not support dynamically splitting and locking the processor.

Type: uint32\_t

Default value: 0

**MFILTEREN**

Enables filtering of address ranges.

Type: bool

Default value: false

**MFILTEREND**

Specifies the end address for address filtering.

Type: uint32\_t

Default value: 0x0

**MFILTERSTART**

Specifies the start address for address filtering.

Type: uint32\_t

Default value: 0x0

**NUM\_MPU\_REGION**

Sets the number of MPU regions.

Type: uint32\_t

Default value: 12

**PERIPHBASE**

Base address of peripheral memory space.

Type: uint32\_t

Default value: 0xAE000000

**PFILTEREND**

Specifies the end address for peripheral port address filtering.

Type: uint32\_t

Default value: 0x0

**PFILTERSTART**

Specifies the start address for peripheral port address filtering.

Type: uint32\_t

Default value: 0xFFF00000

**cpi\_div**

divider for calculating CPI (Cycles Per Instruction).

Type: uint32\_t

Default value: 1

**cpi\_mul**

multiplier for calculating CPI (Cycles Per Instruction).

Type: uint32\_t

Default value: 1

**dcache-state\_modelled**

Set whether D-cache has stateful implementation.

Type: `bool`

Default value: `false`

**dic-spi\_count**

Number of shared peripheral interrupts implemented.

Type: `uint32_t`

Default value: `0x40`

**ecc\_on**

Enable Error Correcting Code.

Type: `bool`

Default value: `false`

**icache-state\_modelled**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

**reported\_fp\_revision**

Purely cosmetic patch level value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored. Updates the FPSID.revision value.

Type: `int`

Default value: -1

**reported\_patch\_level**

Purely cosmetic patch level value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: `int`

Default value: -1

**reported\_revision\_number**

Purely cosmetic revision number value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: int

Default value: -1

## 3.98 ARMCortexX1CCT

Defined in `LISA/ARMCortexX1CCT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following parameters were added:

- `has_delayed_dbgreg`
- `has_delayed_sysreg`

### About ARMCortexX1CCT

The model supports the following features:

- DynamIQ (DSU) system registers.
- Per-core L2 cache.
- A P-Channel for the cluster and for each core.
- Optional peripheral port.
- L3 cache partition.
- Per-core clock.
- `BROADCASTPERSIST` pin.

The model does not support the following features:

- `BROADCASTCACHEMAINTPOU` pin.
- `COREINSTRRET`, `COREINSTRRUN`, or `nPMBIRQ` signals.
- DynamIQ features that are irrelevant to the programmers' view simulation, for example:
  - Automatic CPU retention mode.
  - Level-3 Cache RAM retention.

- 256-bit wide output transactions.
- ECC and parity schemes are hardware-specific so are not supported.
- Self-test features (MBIST).
- Latency configuration.
- Snoop filtering.
- Cache stashing capability.
- Each thread has its own L1Cache and L2Cache instead of a common cache that is shared by all threads of the core.
- Per-thread parameters are not yet supported but signals are implemented.

**Note**

The `cfgsdisable` signal will be removed in a future release.

This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (`crypto.dll` or `crypto.so`) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

### Iris and MTI instances for ARMCortexX1CCT

This model has the following Iris instances:

Name	Instance type
ARMCortexX1CCT	Cluster_ARM_Cortex-X1C
ARMCortexX1CCT.AMU	PVBusLogger
ARMCortexX1CCT.AMU.mapper	PVBusMapper
ARMCortexX1CCT.DAP	PVBusLogger
ARMCortexX1CCT.DAP.mapper	PVBusMapper
ARMCortexX1CCT.DSU	DSU
ARMCortexX1CCT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexX1CCT.DSU.mpam_busslave	PVBusSlave
ARMCortexX1CCT.DSU.shared_cache	PVCache
ARMCortexX1CCT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexX1CCT.DSU.shared_cache.upstream[Y] (where Y = 0-3)	PVBusSlave
ARMCortexX1CCT.MMAP	PVBusLogger
ARMCortexX1CCT.MMAP.mapper	PVBusMapper
ARMCortexX1CCT.RAS	PVBusLogger
ARMCortexX1CCT.RAS.mapper	PVBusMapper
ARMCortexX1CCT.cpu0	ARM_Cortex-X1C
ARMCortexX1CCT.cpu0.UTLB	TLB
ARMCortexX1CCT.cpu0.debug_rom	debug_rom



Name	Instance type
ARMCortexX1CCT.cpu0.dtlb	TLB
ARMCortexX1CCT.cpu0.l1dcache	PVCache
ARMCortexX1CCT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexX1CCT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexX1CCT.cpu0.l1icache	PVCache
ARMCortexX1CCT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexX1CCT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexX1CCT.cpu0.l2cache	PVCache
ARMCortexX1CCT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexX1CCT.cpu0.l2cache.upstream[U] (where U = 0-1)	PVBusSlave
ARMCortexX1CCT.ext_bus	PVBusLogger
ARMCortexX1CCT.ext_bus.mapper	PVBusMapper
ARMCortexX1CCT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexX1CCT.global_debug_rom	debug_rom
ARMCortexX1CCT.secondary_debug_rom	debug_rom

This model has the following MTI trace components:

Name	Component type
ARMCortexX1CCT.AMU	PVBusLogger
ARMCortexX1CCT.AMU.mapper	PVBusMapper
ARMCortexX1CCT.DAP	PVBusLogger
ARMCortexX1CCT.DAP.mapper	PVBusMapper
ARMCortexX1CCT.DSU	DSU
ARMCortexX1CCT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexX1CCT.DSU.mpam_busslave	PVBusSlave
ARMCortexX1CCT.DSU.shared_cache	PVCache
ARMCortexX1CCT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexX1CCT.DSU.shared_cache.upstream[Y] (where Y = 0-3)	PVBusSlave
ARMCortexX1CCT.MMAP	PVBusLogger
ARMCortexX1CCT.MMAP.mapper	PVBusMapper
ARMCortexX1CCT.RAS	PVBusLogger
ARMCortexX1CCT.RAS.mapper	PVBusMapper
ARMCortexX1CCT.cpu0	ARM_Cortex-X1C
ARMCortexX1CCT.cpu0.UTLB	TLB
ARMCortexX1CCT.cpu0.l1dcache	PVCache
ARMCortexX1CCT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexX1CCT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexX1CCT.cpu0.l1icache	PVCache
ARMCortexX1CCT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexX1CCT.cpu0.l1icache.upstream[0]	PVBusSlave

Name	Component type
ARMCortexX1CCT.cpu0.l2cache	PVCache
ARMCortexX1CCT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexX1CCT.cpu0.l2cache.upstream[U] (where $U = 0-1$ )	PVBusSlave
ARMCortexX1CCT.ext_bus	PVBusLogger
ARMCortexX1CCT.ext_bus.mapper	PVBusMapper
ARMCortexX1CCT.gic_cpuif_decoder_cluster	GLCv3CPUInterfaceDecoder

## Ports for ARMCortexX1CCT

Port	Direction	Protocol	Description
acp_s	slave	PVBus	AXI ACP slave port.
AENDMP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
ASTARTMP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	slave	Signal	CHI defined pins.
broadcastcachemaint	slave	Signal	ACE defined pins.
broadcastouter	slave	Signal	ACE defined pins.
broadcastpersist	slave	Signal	CHI defined pins.
cfgend	slave	Signal	This signal if for EE bit initialisation.
cfgsdisable	slave	Signal	This signal disables write access to some secure Interrupt Controller registers.
cfgte	slave	Signal	This signal provides default exception handling state.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
clusterid	slave	Value	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set ManagerID64 of the core.
clusterpmuirq	master	Signal	DynamlQ pmu irq
CNTHPIRQ	master	Signal	Timer signals to SOC.
CNTHVIRQ	master	Signal	Timer signals to SOC.
CNTPNSIRQ	master	Signal	Timer signals to SOC.
CNTPSIRQ	master	Signal	Timer signals to SOC.
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.
CNTVIRQ	master	Signal	Timer signals to SOC.
commirq	master	Signal	Interrupt signal from debug communications channel.

Port	Direction	Protocol	Description
core_clk_in	slave	ClockSignal	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
cpuporeset	slave	Signal	CPU power on reset. Initializes all the processor logic, including debug logic.
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
ctidbgirq	master	Signal	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	slave	Signal	External debug interface.
dbgnopwrdown	master	Signal	No power-down request.
dbgpwrupreq	master	Signal	Debug power up request.
dev_debug_s	slave	PVBus	External debug interface.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	slave	PVBus	External debug interface.
niden	slave	Signal	External debug interface.
pchannel_cluster	slave	PChannel	PChannel for cluster.
pchannel_core	slave	PChannel	PChannels for cores
pmbirq	master	Signal	Interrupt signal from the statistical profiling unit.
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	master	PVBus	The core will generate bus requests on this port.
pvbus_periph_m	master	PVBus	The core can generate peripheral bus request on this port.
rei	slave	Signal	Per core RAM Error Interrupt.
reset	slave	Signal	Raising this signal will put the core into reset mode.
rvbaraddr	slave	Value_64	Reset vector base address.
sci_m	master	SystemCoherencyInterface	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei	slave	Signal	Per core System Error physical pins.
spiden	slave	Signal	External debug interface.
spniden	slave	Signal	External debug interface.
sporeset	slave	Signal	A single cluster-wide power on reset signal for all resettable registers in DynamIQ.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfiq	slave	Signal	Virtualised FIQ.

Port	Direction	Protocol	Description
vinithi	slave	Signal	This signal controls of the location of the exception vectors at reset.
virq	slave	Signal	Virtualised IRQ.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Per core virtual System Error physical pins.

## Parameters for ARMCortexX1CCT

### **cpuX.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **cpuX.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **cpuX.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **cpuX.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **cpuX.VINITHI**

Reset value of SCTLR.V.

Type: `bool`

Default value: `false`

**cpuX.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter trace\_special\_hlt\_imm16.

Type: bool

Default value: false

**cpuX.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **cpuX.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

### **cpuX.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **cpuX.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **cpuX.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **cpuX.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **cpuX.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**cpuX.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**cpuX.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**cpuX.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**cpuX.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**cpuX.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**cpuX.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**cpuX.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**cpuX.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**cpuX.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**cpuX.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**cpuX.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**cpuX.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**cpuX.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**cpuX.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.



Type: `uint16_t`

Default value: `0xf000`

### **`cpuX.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **BROADCASTATOMIC**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`

Default value: `true`

### **BROADCASTCACHEMAINT**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

Type: `bool`

Default value: `true`

### **BROADCASTOUTER**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

Type: `bool`

Default value: `true`

### **BROADCASTPERSIST**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: `bool`

Default value: `true`

### **CLUSTER\_ID**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels,

the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the ManagerID64 of the core.

Type: `uint32_t`

Default value: `0x0`

### **CMO\_broadcast\_when\_cache\_state\_modelling\_disabled**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. This value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required.

Type: `uint8_t`

Default value: `1`

### **GICDISABLE**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`

Default value: `true`

### **NUM\_CORES**

Number of cores per cluster.

Type: `uint8_t`

Default value: `1`

### **cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

### **cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-maintenance\_latency**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-miss\_latency**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-prefetch\_enabled**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

Type: `bool`

Default value: `false`

**dcache-read\_access\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **dcache-size**

L1 D-Cache size in bytes.

Type: `uint32_t`

Default value: `0x10000`

### **dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **dcache-state\_modelled**

Set whether D-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **dcache-write\_access\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **default\_opmode**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

Type: `uint8_t`

Default value: 4

### **diagnostics**

Enable DynamlQ diagnostic messages.

Type: `bool`

Default value: `false`

### **enable\_simulation\_performance\_optimizations**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of `stage12_tlb_size` parameter to 1024).

Type: `bool`

Default value: `true`

### **ext\_abort\_device\_read\_is\_sync**

Synchronous reporting of device-nGnRE read external aborts.

Type: `bool`

Default value: `false`

### **ext\_abort\_device\_write\_is\_sync**

Synchronous reporting of device-nGnRE write external aborts.

Type: `bool`

Default value: `false`

### **ext\_abort\_so\_read\_is\_sync**

Synchronous reporting of device-nGnRnE read external aborts.

Type: `bool`

Default value: `false`

### **ext\_abort\_so\_write\_is\_sync**

Synchronous reporting of device-nGnRnE write external aborts.

Type: `bool`

Default value: `false`

**force\_zero\_PSTATE\_PAN**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

**has\_acp**

If true, Accelerator Coherency Port is configured.

Type: `bool`

Default value: `false`

**has\_coherent\_icache**

Whether icache invalidation to the point of unification is required for instruction to data coherence. true - Invalidate operations not required.

Type: `bool`

Default value: `false`

**has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**has\_dot\_product**

Possible values of this parameter are:- 1, feature not implemented.- 2, feature is implemented.

Type: `uint8_t`

Default value: 2

**has\_peripheral\_port**

If true, additional AXI peripheral port is configured.

Type: `bool`

Default value: `false`

### **`icache-hit_latency`**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`icache-maintenance_latency`**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`icache-miss_latency`**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`icache-prefetch_enabled`**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

### **`icache-read_access_latency`**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-size**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: `0x10000`

**icache-state\_modelled**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

**invalidate\_code\_cache\_on\_icache\_cmo**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: `uint8_t`

Default value: `0`

**l3cache-hit\_latency**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-maintenance\_latency**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`



**l3cache-miss\_latency**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-read\_access\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size `l3cache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-read\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l3cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-size**

L3 Cache size in bytes.

Type: `uint32_t`

Default value: `0x100000`

**l3cache-snoop\_data\_transfer\_latency**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-snoop\_issue\_latency**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-write_access_latency`**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size `l3cache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-write_latency`**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l3cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`pchannel_treat_simreset_as_poreset`**

Register core as ON state to cluster with simulation reset.

Type: `bool`

Default value: `false`

### **`periph_address_end`**

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

Type: `uint64_t`

Default value: `0x0`

### **`periph_address_start`**

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

Type: `uint64_t`

Default value: `0x0`

### **`ptw_latency`**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

### **tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

### **tlbi\_stall\_enabled**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: bool

Default value: false

### **treat-dcache-cmos-to-pou-as-nop**

Whether dcache invalidation to the point of unification is required for instruction to data coherence. 0 - Invalidate ops required, 1 - Invalidate ops not required and cannot generate faults, 2 - Invalidate ops not required but can generate faults.

Type: uint8\_t

Default value: 0

### **treat\_PAC\_as\_NOP**

Non-architecture parameter. Treat Pointer Authentication as **NOP**. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are **NOP**. Branch, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

Type: bool

Default value: false

### **walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

## 3.99 ARMCortexX1CT

Defined in `LISA/ARMCortexX1CT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following parameters were added:

- `has_delayed_dbgreg`
- `has_delayed_sysreg`

### About ARMCortexX1CT

The model supports the following features:

- DynamIQ (DSU) system registers.
- Per-core L2 cache.
- A P-Channel for the cluster and for each core.
- Optional peripheral port.
- L3 cache partition.
- Per-core clock.
- `BROADCASTPERSIST` pin.

The model does not support the following features:

- `BROADCASTCACHEMAINTPOU` pin.
- `COREINSTRRET`, `COREINSTRRUN`, or `nPMBIRQ` signals.
- DynamIQ features that are irrelevant to the programmers' view simulation, for example:
  - Automatic CPU retention mode.
  - Level-3 Cache RAM retention.
- 256-bit wide output transactions.
- ECC and parity schemes are hardware-specific so are not supported.
- Self-test features (MBIST).
- Latency configuration.
- Snoop filtering.
- Debug power management signals `DBGPWRDUP` and `DBGIRSTREQ` are not implemented, but `DBGPWRUPREQ` and `DBGNOPWRDWN` are implemented.

- Cache stashing capability.
- Each thread has its own L1Cache and L2Cache instead of a common cache that is shared by all threads of the core.
- Per-thread parameters are not yet supported but signals are implemented.

This model has a variable number of cores per cluster, specified using the `NUM_CORES` parameter.

The per-core parameters are preceded by `cpu0.`, where `n` identifies the core (0-3).

This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (`crypto.dll` or `crypto.so`) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

### Iris and MTI instances for ARM CortexX1CT

This model has the following Iris instances:

Name	Instance type
ARM CortexX1CT	Cluster_ARM_Cortex-X1
ARM CortexX1CT.AMU	PVBusLogger
ARM CortexX1CT.AMU.mapper	PVBusMapper
ARM CortexX1CT.DAP	PVBusLogger
ARM CortexX1CT.DAP.mapper	PVBusMapper
ARM CortexX1CT.DSU	DSU
ARM CortexX1CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARM CortexX1CT.DSU.mpam_busslave	PVBusSlave
ARM CortexX1CT.DSU.shared_cache	PVCache
ARM CortexX1CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARM CortexX1CT.DSU.shared_cache.upstream[Y] (where Y = 0-3)	PVBusSlave
ARM CortexX1CT.MMAP	PVBusLogger
ARM CortexX1CT.MMAP.mapper	PVBusMapper
ARM CortexX1CT.RAS	PVBusLogger
ARM CortexX1CT.RAS.mapper	PVBusMapper
ARM CortexX1CT.cpu0	ARM_Cortex-X1
ARM CortexX1CT.cpu0.UTLB	TLB
ARM CortexX1CT.cpu0.debug_rom	debug_rom
ARM CortexX1CT.cpu0.dtlb	TLB
ARM CortexX1CT.cpu0.l1dcache	PVCache
ARM CortexX1CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARM CortexX1CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARM CortexX1CT.cpu0.l1icache	PVCache
ARM CortexX1CT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARM CortexX1CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARM CortexX1CT.cpu0.l2cache	PVCache

Name	Instance type
ARMCortexX1CT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexX1CT.cpu0.l2cache.upstream[U] (where U = 0-1)	PVBusSlave
ARMCortexX1CT.ext_bus	PVBusLogger
ARMCortexX1CT.ext_bus.mapper	PVBusMapper
ARMCortexX1CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexX1CT.global_debug_rom	debug_rom
ARMCortexX1CT.secondary_debug_rom	debug_rom

This model has the following MTI trace components:

Name	Component type
ARMCortexX1CT.AMU	PVBusLogger
ARMCortexX1CT.AMU.mapper	PVBusMapper
ARMCortexX1CT.DAP	PVBusLogger
ARMCortexX1CT.DAP.mapper	PVBusMapper
ARMCortexX1CT.DSU	DSU
ARMCortexX1CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexX1CT.DSU.mpam_busslave	PVBusSlave
ARMCortexX1CT.DSU.shared_cache	PVCache
ARMCortexX1CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexX1CT.DSU.shared_cache.upstream[Y] (where Y = 0-3)	PVBusSlave
ARMCortexX1CT.MMAP	PVBusLogger
ARMCortexX1CT.MMAP.mapper	PVBusMapper
ARMCortexX1CT.RAS	PVBusLogger
ARMCortexX1CT.RAS.mapper	PVBusMapper
ARMCortexX1CT.cpu0	ARM_Cortex-X1
ARMCortexX1CT.cpu0.UTLB	TLB
ARMCortexX1CT.cpu0.l1dcache	PVCache
ARMCortexX1CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexX1CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexX1CT.cpu0.l1licache	PVCache
ARMCortexX1CT.cpu0.l1licache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexX1CT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexX1CT.cpu0.l2cache	PVCache
ARMCortexX1CT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexX1CT.cpu0.l2cache.upstream[U] (where U = 0-1)	PVBusSlave
ARMCortexX1CT.ext_bus	PVBusLogger
ARMCortexX1CT.ext_bus.mapper	PVBusMapper
ARMCortexX1CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

## Ports for ARMCortexX1CT

Port	Direction	Protocol	Description
acp_s	slave	PVBus	AXI ACP slave port.
AENDMP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
ASTARTMP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	slave	Signal	CHI defined pins.
broadcastcachemaint	slave	Signal	ACE defined pins.
broadcastouter	slave	Signal	ACE defined pins.
broadcastpersist	slave	Signal	CHI defined pins.
cfgend	slave	Signal	This signal is for EE bit initialisation.
cfgsdisable	slave	Signal	This signal disables write access to some secure Interrupt Controller registers.
cfgte	slave	Signal	This signal provides default exception handling state.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
clusterid	slave	Value	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set ManagerID64 of the core.
clusterpmuirq	master	Signal	DynamiQ pmu irq
CNTHPIRQ	master	Signal	Timer signals to SOC.
CNTHVIRQ	master	Signal	Timer signals to SOC.
CNTPNSIRQ	master	Signal	Timer signals to SOC.
CNTPSIRQ	master	Signal	Timer signals to SOC.
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.
CNTVIRQ	master	Signal	Timer signals to SOC.
commirq	master	Signal	Interrupt signal from debug communications channel.
core_clk_in	slave	ClockSignal	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
cpuporeset	slave	Signal	CPU power on reset. Initializes all the processor logic, including debug logic.
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
ctidbgirq	master	Signal	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	slave	Signal	External debug interface.

Port	Direction	Protocol	Description
dbgnopwrdown	master	Signal	No power-down request.
dbgpwrupreq	master	Signal	Debug power up request.
dev_debug_s	slave	PVBus	External debug interface.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	slave	PVBus	External debug interface.
niden	slave	Signal	External debug interface.
pchannel_cluster	slave	PChannel	PChannel for cluster.
pchannel_core	slave	PChannel	PChannels for cores
pmbirq	master	Signal	Interrupt signal from the statistical profiling unit.
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	master	PVBus	The core will generate bus requests on this port.
pvbus_periph_m	master	PVBus	The core can generate peripheral bus request on this port.
rei	slave	Signal	Per core RAM Error Interrupt.
reset	slave	Signal	Raising this signal will put the core into reset mode.
rvbaraddr	slave	Value_64	Reset vector base address.
sci_m	master	SystemCoherencyInterface	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei	slave	Signal	Per core System Error physical pins.
spiden	slave	Signal	External debug interface.
spniden	slave	Signal	External debug interface.
sporeset	slave	Signal	A single cluster-wide power on reset signal for all resettable registers in DynamIQ.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfiq	slave	Signal	Virtualised FIQ.
vinithi	slave	Signal	This signal controls of the location of the exception vectors at reset.
virq	slave	Signal	Virtualised IRQ.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Per core virtual System Error physical pins.



## Parameters for ARMCortexX1CT

### **cpuX.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **cpuX.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **cpuX.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **cpuX.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **cpuX.VINITHI**

Reset value of SCTLR.V.

Type: `bool`

Default value: `false`

### **cpuX.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**cpuX.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**cpuX.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**cpuX.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**cpuX.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**cpuX.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**cpuX.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**cpuX.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**cpuX.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**cpuX.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**cpuX.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**cpuX.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: true

**cpuX.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**cpuX.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**cpuX.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**cpuX.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**cpuX.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**cpuX.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**BROADCASTATOMIC**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`

Default value: `true`

**BROADCASTCACHEMAINT**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

Type: `bool`

Default value: `true`

**BROADCASTOUTER**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

Type: `bool`

Default value: `true`

**BROADCASTPERSIST**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: `bool`

Default value: `true`

**CLUSTER\_ID**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the ManagerID64 of the core.

Type: `uint32_t`

Default value: `0x0`

### **CMO\_broadcast\_when\_cache\_state\_modelling\_disabled**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. This value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required.

Type: `uint8_t`

Default value: `1`

### **GICDISABLE**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`

Default value: `true`

### **NUM\_CORES**

Number of cores per cluster.

Type: `uint8_t`

Default value: `1`

### **cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

### **cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

### **dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`dcache-maintenance_latency`**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`dcache-miss_latency`**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`dcache-prefetch_enabled`**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

Type: `bool`

Default value: `false`

### **`dcache-read_access_latency`**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`dcache-read_latency`**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`



**dcache-size**

L1 D-Cache size in bytes.

Type: `uint32_t`

Default value: `0x10000`

**dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-state\_modelled**

Set whether D-cache has stateful implementation.

Type: `bool`

Default value: `false`

**dcache-write\_access\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**default\_opmode**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

Type: `uint8_t`

Default value: 4

**diagnostics**

Enable DynaMiq diagnostic messages.

Type: `bool`

Default value: `false`

**enable\_simulation\_performance\_optimizations**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of `stage12_tlb_size` parameter to 1024).

Type: `bool`

Default value: `true`

**ext\_abort\_device\_read\_is\_sync**

Synchronous reporting of device-nGnRE read external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_device\_write\_is\_sync**

Synchronous reporting of device-nGnRE write external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_so\_read\_is\_sync**

Synchronous reporting of device-nGnRnE read external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_so\_write\_is\_sync**

Synchronous reporting of device-nGnRnE write external aborts.

Type: `bool`

Default value: `false`

**force\_zero\_PSTATE\_PAN**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

**has\_acp**

If true, Accelerator Coherency Port is configured.

Type: `bool`

Default value: `false`

**has\_coherent\_icache**

Whether icache invalidation to the point of unification is required for instruction to data coherence.  
`true` - Invalidate operations not required.

Type: `bool`

Default value: `false`

**has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**has\_dot\_product**

Possible values of this parameter are:- 1, feature not implemented.- 2, feature is implemented.

Type: `uint8_t`

Default value: 2

**has\_peripheral\_port**

If true, additional AXI peripheral port is configured.

Type: `bool`

Default value: `false`

### **`icache-hit_latency`**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`icache-maintenance_latency`**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`icache-miss_latency`**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`icache-prefetch_enabled`**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

### **`icache-read_access_latency`**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-size**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: `0x10000`

**icache-state\_modelled**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

**invalidate\_code\_cache\_on\_icache\_cmo**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: `uint8_t`

Default value: `0`

**l3cache-hit\_latency**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-maintenance\_latency**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-miss\_latency**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-read\_access\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size `l3cache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-read\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l3cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-size**

L3 Cache size in bytes.

Type: `uint32_t`

Default value: `0x100000`

**l3cache-snoop\_data\_transfer\_latency**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-snoop\_issue\_latency**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-write_access_latency`**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size `l3cache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-write_latency`**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l3cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`pchannel_treat_simreset_as_poreset`**

Register core as ON state to cluster with simulation reset.

Type: `bool`

Default value: `false`

### **`periph_address_end`**

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

Type: `uint64_t`

Default value: `0x0`

### **`periph_address_start`**

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

Type: `uint64_t`

Default value: `0x0`

### **`ptw_latency`**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

### **tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

### **tlbi\_stall\_enabled**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: bool

Default value: false

### **treat-dcache-cmos-to-pou-as-nop**

Whether dcache invalidation to the point of unification is required for instruction to data coherence. 0 - Invalidate ops required, 1 - Invalidate ops not required and cannot generate faults, 2 - Invalidate ops not required but can generate faults.

Type: uint8\_t

Default value: 0

### **walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

## 3.100 ARMCortexX2CT

Defined in LISA/ARMCortexX2CT.lisa.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r2p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following parameters were added:



- `has_delayed_dbgreg`
- `has_delayed_sysreg`
- `num_acp`

## About ARMCortexX2CT

The model supports the following features:

- DynamIQ (DSU) system registers.
- Per-core L2 cache.
- A P-Channel for the cluster and for each core.
- Optional peripheral port.
- L3 cache partition.
- Per-core clock.
- `BROADCASTPERSIST` pin.

Support for the following features is planned for a future release:

- `BROADCASTCACHEMAINTPOU` pin.
- `COREINSTRRET`, `COREINSTRRUN`, and `nPMBIRQ` signals.

The following features will not be implemented:

- 256-bit wide output transactions.
- Error correction/detection features.
- Self-test features (MBIST).
- Snoop filtering.

This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (`crypto.dll` or `crypto.so`) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

## Iris and MTI instances for ARMCortexX2CT

This model has the following Iris instances:

Name	Instance type
ARMCortexX2CT	Cluster_ARM_Cortex-X2
ARMCortexX2CT.AMU	PVBusLogger
ARMCortexX2CT.AMU.mapper	PVBusMapper
ARMCortexX2CT.DAP	PVBusLogger
ARMCortexX2CT.DAP.mapper	PVBusMapper
ARMCortexX2CT.DSU	DSU-110
ARMCortexX2CT.DSU.PPU_cluster	PPUv1
ARMCortexX2CT.DSU.PPU_cluster.busslave	PVBusSlave

Name	Instance type
ARMCortexX2CT.DSU.PPU_core0	PPUv1
ARMCortexX2CT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexX2CT.DSU.mpam_busslave	PVBusSlave
ARMCortexX2CT.DSU.shared_cache	PVCache
ARMCortexX2CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexX2CT.DSU.shared_cache.upstream[Y] (where Y = 0-3)	PVBusSlave
ARMCortexX2CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexX2CT.MMAP	PVBusLogger
ARMCortexX2CT.MMAP.mapper	PVBusMapper
ARMCortexX2CT.RAS	PVBusLogger
ARMCortexX2CT.RAS.mapper	PVBusMapper
ARMCortexX2CT.cpu0	ARM_Cortex-X2
ARMCortexX2CT.cpu0.UTLB	TLB
ARMCortexX2CT.cpu0.debug_rom	debug_rom
ARMCortexX2CT.cpu0.dtlb	TLB
ARMCortexX2CT.cpu0.l1dcache	PVCache
ARMCortexX2CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexX2CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexX2CT.cpu0.l1icache	PVCache
ARMCortexX2CT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexX2CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexX2CT.cpu0.l2cache	PVCache
ARMCortexX2CT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexX2CT.cpu0.l2cache.upstream[U] (where U = 0-1)	PVBusSlave
ARMCortexX2CT.ext_bus	PVBusLogger
ARMCortexX2CT.ext_bus.mapper	PVBusMapper
ARMCortexX2CT.gic_cpuif_decoder_cluster	GIcV3CPUInterfaceDecoder
ARMCortexX2CT.global_debug_rom	debug_rom
ARMCortexX2CT.secondary_debug_rom	debug_rom
ARMCortexX2CT.sve	ScalableVectorExtension

This model has the following MTI trace components:

Name	Component type
ARMCortexX2CT.AMU	PVBusLogger
ARMCortexX2CT.AMU.mapper	PVBusMapper
ARMCortexX2CT.DAP	PVBusLogger
ARMCortexX2CT.DAP.mapper	PVBusMapper
ARMCortexX2CT.DSU	DSU-110
ARMCortexX2CT.DSU.PPU_cluster	PPUv1
ARMCortexX2CT.DSU.PPU_cluster.busslave	PVBusSlave

Name	Component type
ARMCortexX2CT.DSU.PPU_core0	PPUv1
ARMCortexX2CT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexX2CT.DSU.mpam_busslave	PVBusSlave
ARMCortexX2CT.DSU.shared_cache	PVCache
ARMCortexX2CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexX2CT.DSU.shared_cache.upstream[Y] (where Y = 0-3)	PVBusSlave
ARMCortexX2CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexX2CT.MMAP	PVBusLogger
ARMCortexX2CT.MMAP.mapper	PVBusMapper
ARMCortexX2CT.RAS	PVBusLogger
ARMCortexX2CT.RAS.mapper	PVBusMapper
ARMCortexX2CT.cpu0	ARM_Cortex-X2
ARMCortexX2CT.cpu0.UTLB	TLB
ARMCortexX2CT.cpu0.l1dcache	PVCache
ARMCortexX2CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexX2CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexX2CT.cpu0.l1icache	PVCache
ARMCortexX2CT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexX2CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexX2CT.cpu0.l2cache	PVCache
ARMCortexX2CT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexX2CT.cpu0.l2cache.upstream[U] (where U = 0-1)	PVBusSlave
ARMCortexX2CT.ext_bus	PVBusLogger
ARMCortexX2CT.ext_bus.mapper	PVBusMapper
ARMCortexX2CT.gic_cpuif_decoder_cluster	GLCv3CPUInterfaceDecoder

## Ports for ARMCortexX2CT

Port	Direction	Protocol	Description
acp_s	slave	PVBus	AXI ACP slave port.
AEND0MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND2MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND3MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).

Port	Direction	Protocol	Description
ASTART2MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	slave	Signal	CHI defined pins.
broadcastcachemaint	slave	Signal	ACE defined pins.
broadcastouter	slave	Signal	ACE defined pins.
broadcastpersist	slave	Signal	CHI defined pins.
cfgend	slave	Signal	This signal is for EE bit initialisation.
cfgsdisable	slave	Signal	This signal disables write access to some secure Interrupt Controller registers.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcs_m_pchannel	master	PChannel	Cluster PCSM signal
cluster_powerdown_out	master	Signal	DEPRECATED - An output from PPUs that informs thermal controller of the cluster power info
cluster_ppu_hw_stat	master	Value	Cluster PPU power state
clusterid	slave	Value	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set ManagerID64 of the core.
clusterpmuirq	master	Signal	DynamlQ pmu irq
CNTHPIRQ	master	Signal	Timer signals to SOC.
CNTHPSIRQ	master	Signal	Timer signals to SOC
CNTHVIRQ	master	Signal	Timer signals to SOC.
CNTHVSIRQ	master	Signal	Timer signals to SOC
CNTPNSIRQ	master	Signal	Timer signals to SOC.
CNTPSIRQ	master	Signal	Timer signals to SOC.
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.
CNTVIRQ	master	Signal	Timer signals to SOC.
commirq	master	Signal	Interrupt signal from debug communications channel.
core_clk_in	slave	ClockSignal	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
core_pcs_m_pchannel	master	PChannel	Core PCSM signals
core_powerdown_out	master	Signal	DEPRECATED - An output from PPUs that informs thermal controller of the core power info

Port	Direction	Protocol	Description
core_ppu_hw_stat	master	Value	Notify the power state change inside the PPU.
coreerrirq	master	Signal	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq	master	Signal	Fault indicator for a detected 1 or 2 bit ECC error
coreinstrrun	master	Signal	Core Running State
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
ctidbgirq	master	Signal	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	slave	Signal	External debug interface.
dbgnopwrdown	master	Signal	No power-down request.
dbgprupreq	master	Signal	Debug power up request.
defaultMP	slave	Signal	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main master interface ports.
dev_debug_s	slave	PVBus	External debug interface.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.
gicreset	master	Signal	An output from PPU that can be used to reset GICCLK domain components
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	slave	PVBus	External debug interface.
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	master	Signal	PPU cluster interrupt.
ppu_cluster_isolate	master	Signal	PPU Cluster Domain isolation control.
ppu_cluster_wakerequest	slave	Signal	PPU cluster wake request signal.
ppu_core_irq	master	Signal	PPU core interrupt.
ppu_core_isolate	master	Signal	PPU Core Domain isolation control.
ppu_core_wakerequest	slave	Signal	PPU Core wake request signals.
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	master	PVBus	The core will generate bus requests on this port.
pvbus_periph_m	master	PVBus	The core can generate peripheral bus request on this port.
rei	slave	Signal	Per core RAM Error Interrupt.
reset	slave	Signal	Reset signal to cluster.
rvbaraddr	slave	Value_64	Reset vector base address.
sci_m	master	SystemCoherencyInterface	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei	slave	Signal	Per core System Error physical pins.

Port	Direction	Protocol	Description
spiden	slave	Signal	External debug interface.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq	master	Signal	Interrupt signal from the trace buffer unit.
utility_bus_s	slave	PVBus	Utility bus slave
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfiq	slave	Signal	Virtualised FIQ.
virq	slave	Signal	Virtualised IRQ.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Per core virtual System Error physical pins.

## Parameters for ARMCortexX2CT

### **cpuX.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **cpuX.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **cpuX.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **cpuX.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**cpuX.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

### **cpuX.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **cpuX.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **cpuX.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **cpuX.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **cpuX.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

### **cpuX.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t



Default value: 0

**cpuX.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**cpuX.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**cpuX.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**cpuX.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**cpuX.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**cpuX.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**cpuX.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**cpuX.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: true

**cpuX.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**cpuX.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**cpuX.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**cpuX.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**cpuX.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**cpuX.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**BROADCASTATOMIC**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`

Default value: `true`

**BROADCASTCACHEMAINT**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

Type: `bool`

Default value: `true`

**BROADCASTOUTER**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

Type: `bool`

Default value: `true`

**BROADCASTPERSIST**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: `bool`

Default value: `true`

**CLUSTER\_ID**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the ManagerID64 of the core.

Type: `uint32_t`

Default value: `0x0`

### **CMO\_broadcast\_when\_cache\_state\_modelling\_disabled**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. This value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required.

Type: `uint8_t`

Default value: `1`

### **GICDISABLE**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`

Default value: `true`

### **NUM\_CORES**

Number of cores per cluster.

Type: `uint8_t`

Default value: `1`

### **core\_power\_on\_by\_default**

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPU which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

Type: `bool`

Default value: `false`

### **cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**`cpi_mul`**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**`dcache-hit_latency`**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**`dcache-maintenance_latency`**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**`dcache-miss_latency`**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**`dcache-prefetch_enabled`**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

Type: `bool`

Default value: `false`

**`dcache-read_access_latency`**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-state\_modelled**

Set whether D-cache has stateful implementation.

Type: `bool`

Default value: `false`

**dcache-write\_access\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**default\_opmode**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

Type: `uint8_t`

Default value: 4

### **diagnostics**

Enable DynamlQ diagnostic messages.

Type: `bool`

Default value: `false`

### **enable\_simulation\_performance\_optimizations**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of `stage12_tlb_size` parameter to 1024).

Type: `bool`

Default value: `true`

### **ete.CLAIMTAGS**

Number of claim tags.

Type: `uint8_t`

Default value: 32

### **ete.MAX\_INST\_PER\_Q**

Maximum limit for the number of instructions implied by a Q element.

Type: `uint16_t`

Default value: `0x1`

### **ete.NumberOfRSPairs**

Number of resource selector pairs.

Type: `uint8_t`

Default value: 8

### **ete.PIDR\_CM0D**

TRCPIDR CM0D value.

Type: `uint8_t`

Default value: 0

**ete.PIDR\_REVAND**

TRCPIDR REVAND value.

Type: `uint8_t`

Default value: 0

**ete.PIDR\_REVISION**

TRCPIDR REVISION value.

Type: `uint8_t`

Default value: 0

**ete.Q\_CADENCE**

Number of instruction blocks traced between two Q elements.

Type: `uint16_t`

Default value: 0x1

**ete.RES0\_STATEFUL**

Whether **RES0** bits are stateful or **RAZ/WI**.

Type: `bool`

Default value: `false`

**ete.RETSTACK**

Return stack depth.

Type: `uint8_t`

Default value: 3

**ete.REVISION**

TRCIDR1 revision value.

Type: `uint8_t`

Default value: 2

**ete.SIM\_OVERFLOW\_GRANULARITY**

Number of instruction blocks in each granule, for simulated overflow.

Type: `uint32_t`

Default value: 0x64



**ete.SIM\_OVERFLOW\_PERCENTAGE**

Percentage of instruction blocks lost in each granule, for simulated overflow.

Type: `uint8_t`

Default value: `0`

**ete.SOURCE\_ADDRESS**

Allow generation of source address elements.

Type: `bool`

Default value: `false`

**ete.TRACE\_OUTPUT**

File to which to write trace byte stream.

Type: `string`

Default value: `N/A`

**ete.TRCSRSTA\_FORCED\_EXCEP**

TRCSRSTA value for a forcibly traced exception.

Type: `bool`

Default value: `false`

**ext\_abort\_device\_read\_is\_sync**

Synchronous reporting of device-nGnRE read external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_device\_write\_is\_sync**

Synchronous reporting of device-nGnRE write external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_so\_read\_is\_sync**

Synchronous reporting of device-nGnRnE read external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_so\_write\_is\_sync**

Synchronous reporting of device-nGnRnE write external aborts.

Type: `bool`

Default value: `false`

**force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks**

Force MTE tag accesses to **RAZ/WI** and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

Type: `bool`

Default value: `false`

**force\_zero\_PSTATE\_PAN**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

**force\_zero\_mpam\_partid\_and\_pmg**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAMO\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

Type: `uint8_t`

Default value: 0

**has\_acp**

If true, Accelerator Coherency Port is configured.

Type: `bool`

Default value: `false`

**has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**has\_ete**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (-plugin or -P).

Type: `bool`

Default value: `false`

**has\_peripheral\_port**

If true, an additional AXI peripheral port is configured.

Type: `bool`

Default value: `false`

**icache-hit\_latency**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

Type: `bool`

Default value: `false`

**icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when icache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**icache-state\_modelled**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

**invalidate\_code\_cache\_on\_icache\_cmo**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: `uint8_t`

Default value: `0`

**l3cache-hit\_latency**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

### **l3cache-maintenance\_latency**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **l3cache-miss\_latency**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **l3cache-read\_access\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size `l3cache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **l3cache-read\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l3cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **l3cache-size**

L3 Cache size in bytes.

Type: `uint32_t`

Default value: `0x0`

**l3cache-snoop\_data\_transfer\_latency**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-snoop\_issue\_latency**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-write\_access\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size `l3cache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-write\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l3cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**memory\_tagging\_support\_level**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).

Type: `uint8_t`

Default value: 2

**mpmm\_accumulator\_multiplier**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator.value of `n` means the accumulator will use  $(n * \text{accumulator value})$  to calculate the mpmm threshold (MPMM).is provided as a fast model workaround to handle cases where execution of quantums

in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

Type: `uint8_t`

Default value: 1

### **num\_acp**

Number of ACP ports.

Type: `uint8_t`

Default value: 0

### **num\_nodes**

Number of transport nodes. Zero implies direct-connect configuration.

Type: `uint8_t`

Default value: 1

### **periph\_address\_end**

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

Type: `uint64_t`

Default value: 0x0

### **periph\_address\_start**

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

Type: `uint64_t`

Default value: 0x0

### **ptw\_latency**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

### **tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

**tlbi\_stall\_enabled**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: `bool`

Default value: `false`

**treat-dcache-cmos-to-pou-as-nop**

Whether dcache invalidation to the point of unification is required for instruction to data coherence. 0 - Invalidate ops required, 1 - Invalidate ops not required and cannot generate faults, 2 - Invalidate ops not required but can generate faults.

Type: `uint8_t`

Default value: 0

**walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

## 3.101 ARMCortexX3CT

Defined in `LISA/ARMCortexX3CT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following parameters were added:

- `has_delayed_dbgreg`
- `has_delayed_sysreg`
- `num_acp`

### About ARMCortexX3CT

The model supports the following features:

- DynamIQ (DSU) system registers.



- DynamIQ r3p0.
- L2 cache is supported at the per-core level only and there is no implementation of Core-Complex with shared L2 cache yet.
- P-Channel for the cluster and for each core.
- Optional peripheral port.
- L3 cache partition.
- Per-core clock.
- BROADCASTPERSIST pin.
- Utility bus.
- Support for Interrupt signals from SPE is added as pmbirq[8].

Support for the following features is planned for a future release:

- TRBE.
- Core-Complex.
- BROADCASTCACHEMAINTPOU pin.
- COREINSTRRET, COREINSTRRUN, or nPMBIRQ signals.
- Each thread has its own L1Cache and L2Cache instead of a common cache that is shared by all threads of the core.
- Per-thread parameters are not yet supported but signals are implemented.



The `cfgsdisable` signal will be removed in a future release.

---

The following features will not be implemented:

- DynamIQ features that are irrelevant to the programmers' view simulation, for example:
  - Automatic CPU retention mode.
  - Level-3 Cache RAM retention.
- 256-bit wide output transactions.
- ECC and parity schemes are hardware-specific so are not supported.
- Self-test features (MBIST).
- Latency configuration.
- Snoop filtering.
- Cache stashing capability.
- Embedded Logic Analyzer (ELA).

This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (crypto.dll or crypto.so) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

## Limitations

Some of the SVE instructions do not raise undefined exceptions when SP is used as a source register.

## Iris and MTI instances for ARMCortexX3CT

This model has the following Iris instances:

Name	Instance type
ARMCortexX3CT	Cluster_ARM_Cortex-X3
ARMCortexX3CT.AMU	PVBusLogger
ARMCortexX3CT.AMU.mapper	PVBusMapper
ARMCortexX3CT.DAP	PVBusLogger
ARMCortexX3CT.DAP.mapper	PVBusMapper
ARMCortexX3CT.DSU	DSU-110
ARMCortexX3CT.DSU.PPU_cluster	PPUv1
ARMCortexX3CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexX3CT.DSU.PPU_core0	PPUv1
ARMCortexX3CT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexX3CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexX3CT.DSU.mpam_busslave	PVBusSlave
ARMCortexX3CT.DSU.shared_cache	PVCache
ARMCortexX3CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexX3CT.DSU.shared_cache.upstream[Y] (where Y = 0-3)	PVBusSlave
ARMCortexX3CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexX3CT.MMAP	PVBusLogger
ARMCortexX3CT.MMAP.mapper	PVBusMapper
ARMCortexX3CT.RAS	PVBusLogger
ARMCortexX3CT.RAS.mapper	PVBusMapper
ARMCortexX3CT.cpu0	ARM_Cortex-X3
ARMCortexX3CT.cpu0.UTLB	TLB
ARMCortexX3CT.cpu0.debug_rom	debug_rom
ARMCortexX3CT.cpu0.dtlb	TLB
ARMCortexX3CT.cpu0.l1dcache	PVCache
ARMCortexX3CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexX3CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexX3CT.cpu0.l1icache	PVCache
ARMCortexX3CT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexX3CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexX3CT.cpu0.l2cache	PVCache

Name	Instance type
ARMCortexX3CT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexX3CT.cpu0.l2cache.upstream[U] (where U = 0–1)	PVBusSlave
ARMCortexX3CT.ext_bus	PVBusLogger
ARMCortexX3CT.ext_bus.mapper	PVBusMapper
ARMCortexX3CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexX3CT.global_debug_rom	debug_rom
ARMCortexX3CT.secondary_debug_rom	debug_rom
ARMCortexX3CT.sve	ScalableVectorExtension

This model has the following MTI trace components:

Name	Component type
ARMCortexX3CT.AMU	PVBusLogger
ARMCortexX3CT.AMU.mapper	PVBusMapper
ARMCortexX3CT.DAP	PVBusLogger
ARMCortexX3CT.DAP.mapper	PVBusMapper
ARMCortexX3CT.DSU	DSU-110
ARMCortexX3CT.DSU.PPU_cluster	PPUv1
ARMCortexX3CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexX3CT.DSU.PPU_core0	PPUv1
ARMCortexX3CT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexX3CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexX3CT.DSU.mpam_busslave	PVBusSlave
ARMCortexX3CT.DSU.shared_cache	PVCache
ARMCortexX3CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexX3CT.DSU.shared_cache.upstream[Y] (where Y = 0–3)	PVBusSlave
ARMCortexX3CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexX3CT.MMAP	PVBusLogger
ARMCortexX3CT.MMAP.mapper	PVBusMapper
ARMCortexX3CT.RAS	PVBusLogger
ARMCortexX3CT.RAS.mapper	PVBusMapper
ARMCortexX3CT.cpu0	ARM_Cortex-X3
ARMCortexX3CT.cpu0.UTLB	TLB
ARMCortexX3CT.cpu0.l1dcache	PVCache
ARMCortexX3CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexX3CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexX3CT.cpu0.l1icache	PVCache
ARMCortexX3CT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexX3CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexX3CT.cpu0.l2cache	PVCache
ARMCortexX3CT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster

Name	Component type
ARMCortexX3CT.cpu0.l2cache.upstream[U] (where U = 0-1)	PVBusSlave
ARMCortexX3CT.ext_bus	PVBusLogger
ARMCortexX3CT.ext_bus.mapper	PVBusMapper
ARMCortexX3CT.gic_cpuif_decoder_cluster	GlCv3CPUInterfaceDecoder

## Ports for ARMCortexX3CT

Port	Direction	Protocol	Description
acp_s	slave	PVBus	AXI ACP slave port.
AEND0MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND2MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND3MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART2MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	slave	Signal	CHI defined pins.
broadcastcachemaint	slave	Signal	ACE defined pins.
broadcastouter	slave	Signal	ACE defined pins.
broadcastpersist	slave	Signal	CHI defined pins.
cfgend	slave	Signal	This signal is for EE bit initialisation.
cfgsdisable	slave	Signal	This signal disables write access to some secure Interrupt Controller registers.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcsn_pchannel	master	PChannel	Cluster PCSM signal
cluster_powerdown_out	master	Signal	DEPRECATED - An output from PPUs that informs thermal controller of the cluster power info
cluster_ppu_hw_stat	master	Value	Cluster PPU power state

Port	Direction	Protocol	Description
clusterid	slave	Value	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set ManagerID64 of the core.
clusterpmuirq	master	Signal	DynamiQ pmu irq
CNTHPIRQ	master	Signal	Timer signals to SOC.
CNTHPSIRQ	master	Signal	Timer signals to SOC
CNTHVIRQ	master	Signal	Timer signals to SOC.
CNTHVSIRQ	master	Signal	Timer signals to SOC
CNTPNSIRQ	master	Signal	Timer signals to SOC.
CNTPSIRQ	master	Signal	Timer signals to SOC.
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.
CNTVIRQ	master	Signal	Timer signals to SOC.
commirq	master	Signal	Interrupt signal from debug communications channel.
core_clk_in	slave	ClockSignal	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
core_pcsn_pchannel	master	PChannel	Core PCSM signals
core_powerdown_out	master	Signal	DEPRECATED - An output from PPUs that informs thermal controller of the core power info
core_ppu_hw_stat	master	Value	Notify the power state change inside the PPU.
coreerrirq	master	Signal	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq	master	Signal	Fault indicator for a detected 1 or 2 bit ECC error
coreinstrrun	master	Signal	Core Running State
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
ctidbgirq	master	Signal	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	slave	Signal	External debug interface.
dbgnopwrdown	master	Signal	No power-down request.
dbgprupreq	master	Signal	Debug power up request.
defaultMP	slave	Signal	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main master interface ports.
dev_debug_s	slave	PVBus	External debug interface.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.

Port	Direction	Protocol	Description
gicreset	master	Signal	An output from PPUs that can be used to reset GICCLK domain components
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	slave	PVBus	External debug interface.
pmbirq	master	Signal	Interrupt signal from the statistical profiling unit.
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	master	Signal	PPU cluster interrupt.
ppu_cluster_isolate	master	Signal	PPU Cluster Domain isolation control.
ppu_cluster_wakerequest	slave	Signal	PPU cluster wake request signal.
ppu_core_irq	master	Signal	PPU core interrupt.
ppu_core_isolate	master	Signal	PPU Core Domain isolation control.
ppu_core_wakerequest	slave	Signal	PPU Core wake request signals.
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	master	PVBus	The core will generate bus requests on this port.
pvbus_periph_m	master	PVBus	The core can generate peripheral bus request on this port.
rei	slave	Signal	Per core RAM Error Interrupt.
reset	slave	Signal	Reset signal to cluster.
rvbaraddr	slave	Value_64	Reset vector base address.
sci_m	master	SystemCoherencyInterface	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei	slave	Signal	Per core System Error physical pins.
spiden	slave	Signal	External debug interface.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq	master	Signal	Interrupt signal from the trace buffer unit.
utility_bus_s	slave	PVBus	Utility bus slave
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfiq	slave	Signal	Virtualised FIQ.
virq	slave	Signal	Virtualised IRQ.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Per core virtual System Error physical pins.

## Parameters for ARMCortexX3CT

### **cpuX.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **cpuX.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **cpuX.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **cpuX.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **cpuX.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **cpuX.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**cpuX.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t



Default value: 0x100000

### **cpuX.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **cpuX.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **cpuX.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **cpuX.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **cpuX.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

### **cpuX.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**cpuX.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**cpuX.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**cpuX.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**cpuX.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**cpuX.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**cpuX.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**cpuX.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**cpuX.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: true

**cpuX.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**cpuX.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**cpuX.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**cpuX.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**cpuX.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**cpuX.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**cpuX.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**AEND0\_DEFAULT**

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

Type: `uint64_t`

Default value: `0x0`

**AEND1\_DEFAULT**

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

Type: `uint64_t`

Default value: `0x0`

**AEND2\_DEFAULT**

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

Type: `uint64_t`

Default value: `0x0`

**AEND3\_DEFAULT**

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART0\_DEFAULT**

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART1\_DEFAULT**

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART2\_DEFAULT**

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART3\_DEFAULT**

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

Type: `uint64_t`

Default value: `0x0`

**BROADCASTATOMIC**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`

Default value: `true`

**BROADCASTCACHEMAINT**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

Type: `bool`

Default value: `false`

**BROADCASTOUTER**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

Type: `bool`

Default value: `false`

**BROADCASTPERSIST**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: `bool`

Default value: `true`

**CLUSTER\_ID**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the ManagerID64 of the core.

Type: `uint32_t`

Default value: `0x0`

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. This value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required.

Type: `uint8_t`

Default value: `1`

**CPUCFR**

Value of CPU Configuration Register.

Type: `uint64_t`

Default value: `0x0`

**GICDISABLE**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`

Default value: `true`

**NUM\_CORES**

Number of cores per cluster.

Type: `uint8_t`

Default value: 1

**bus\_type**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

Type: `uint8_t`

Default value: 0

**core\_power\_on\_by\_default**

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPUs which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

Type: `bool`

Default value: `false`

**cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: 0x1

**cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: 0x1

**dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-maintenance\_latency**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-miss\_latency**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-prefetch\_enabled**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

Type: `bool`

Default value: `false`

**dcache-read\_access\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.



Type: `uint64_t`

Default value: `0x0`

### **dcache-size**

L1 D-Cache size in bytes.

Type: `uint32_t`

Default value: `0x8000`

### **dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **dcache-state\_modelled**

Set whether D-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **dcache-write\_access\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **default\_opmode**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

Type: `uint8_t`

Default value: 4

### **diagnostics**

Enable DynamlQ diagnostic messages.

Type: `bool`

Default value: `false`

### **enable\_simulation\_performance\_optimizations**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of `stage12_tlb_size` parameter to 1024).

Type: `bool`

Default value: `true`

### **error\_record\_feature\_register**

RAS feature register values. An array of JSON objects. The JSON schema for the array is: `[{"ED":0x0, "IMPDEF_3_2":0x0, "UI":0x0, "FI":0x0, "UE":0x0, "CFI":0x0, "CEC":0x0, "RP":0x0, "DUI":0x0, "CEO":0x0, "CI":0x0, "TS":0x0, "INJ":0x0, "FRX":0x0, "UC":0x0, "UEU":0x0, "UER":0x0, "UEO":0x0, "DE":0x0, "CE":0x0, "Visibility":"Core"},other_feature_register_values]`. Where ED,UI,FI,CE and UE have valid values between 0x0 - 0x3. CFI and DUI have valid values 0x0, 0x2 and 0x3. CEC has valid values 0x0,0x2 or 0x4. RP,CEO,INJ,FRX,UC,UEU,UER,UEO,DE has valid values 0x0 or 0x1. CI and TS has valid values of 0x0, 0x1 and 0x2. Visibility has valid values "Core" or "Cluster".

Type: `string`

Default value: `"[{\"ED\":0x2,\"IMPDEF_3_2\":0x0,\"UI\":0x2,\"FI\":0x2,\"UE\":0x1,\"CFI\":0x2,\"CEC\":0x2,\"RP\":0x1,\"DUI\":0x0,\"CEO\":0x0,\"INJ\":0x1,\"CI\":0x0,\"TS\":0x0,\"Visibility\": \"Cluster\"},{\"ED\":0x2,\"IMPDEF_3_2\":0x0,\"UI\":0x2,\"FI\":0x2,\"UE\":0x1,\"CFI\":0x2,\"CEC\":0x2,\"RP\":0x1,\"DUI\":0x0,\"CEO\":0x0,\"INJ\":0x1,\"CI\":0x0,\"TS\":0x0}]]\"`

### **ete.CLAIMTAGS**

Number of claim tags.

Type: `uint8_t`

Default value: 32

### **ete.MAX\_INST\_PER\_Q**

Maximum limit for the number of instructions implied by a Q element.

Type: `uint16_t`

Default value: `0x1`

#### **`ete.NumberOfRSPairs`**

Number of resource selector pairs.

Type: `uint8_t`

Default value: `8`

#### **`ete.PIDR_CM0D`**

TRCPIDR CM0D value.

Type: `uint8_t`

Default value: `0`

#### **`ete.PIDR_REVAND`**

TRCPIDR REVAND value.

Type: `uint8_t`

Default value: `0`

#### **`ete.PIDR_REVISION`**

TRCPIDR REVISION value.

Type: `uint8_t`

Default value: `0`

#### **`ete.Q_CADENCE`**

Number of instruction blocks traced between two Q elements.

Type: `uint16_t`

Default value: `0x1`

#### **`ete.RES0_STATEFUL`**

Whether **RES0** bits are stateful or **RAZ/WI**.

Type: `bool`

Default value: `false`

#### **`ete.RETSTACK`**

Return stack depth.

Type: `uint8_t`

Default value: 3

#### **ete.REVISION**

TRCIDR1 revision value.

Type: `uint8_t`

Default value: 0

#### **ete.SIM\_OVERFLOW\_GRANULARITY**

Number of instruction blocks in each granule, for simulated overflow.

Type: `uint32_t`

Default value: 0x64

#### **ete.SIM\_OVERFLOW\_PERCENTAGE**

Percentage of instruction blocks lost in each granule, for simulated overflow.

Type: `uint8_t`

Default value: 0

#### **ete.SOURCE\_ADDRESS**

Allow generation of source address elements.

Type: `bool`

Default value: `false`

#### **ete.TRACE\_OUTPUT**

File to which to write trace byte stream.

Type: `string`

Default value: N/A

#### **ete.TRCRSRTA\_FORCED\_EXCEP**

TRCRSR.TA value for a forcibly traced exception.

Type: `bool`

Default value: `false`

#### **ext\_abort\_normal\_noncacheable\_read\_is\_sync**

Synchronous reporting of normal noncacheable-read external aborts.

Type: `bool`

Default value: `true`

### **ext\_abort\_so\_write\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `uint8_t`

Default value: 2

### **force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks**

Force MTE tag accesses to **RAZ/WI** and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

Type: `bool`

Default value: `false`

### **force\_zero\_PSTATE\_PAN**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

### **force\_zero\_mpam\_partid\_and\_pmg**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

Type: `uint8_t`

Default value: 0

### **has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

### **has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

### **has\_enhanced\_pan**

Implements Armv8.7 Enhanced PAN feature (FEAT\_PAN3) values of this parameter are:- 1, feature is implemented if ARMv8.7 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 2

### **has\_ete**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (-plugin or -P).

Type: `bool`

Default value: `false`

### **has\_large\_va**

Implement support for the extended 52-bit virtual addresses from ARMv8.2 (FEAT\_LVA). values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.2 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **has\_peripheral\_port**

If true, an additional AXI peripheral port is configured.

Type: `bool`

Default value: `false`

### **icache-hit\_latency**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

### **icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **icache-size**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: `0x8000`

**icache-state\_modelled**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

**instruction\_tlb\_size**

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

Type: `uint32_t`

Default value: `0x0`

**invalidate\_code\_cache\_on\_icache\_cmo**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: `uint8_t`

Default value: `0`

**l3cache-has\_mpam**

L3 Cache has MPAM support.

Type: `bool`

Default value: `true`

**l3cache-hit\_latency**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-maintenance\_latency**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`



**l3cache-miss\_latency**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-read\_access\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size `l3cache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-read\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l3cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-size**

L3 Cache size in bytes.

Type: `uint32_t`

Default value: `0x40000`

**l3cache-snoop\_data\_transfer\_latency**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-snoop\_issue\_latency**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-ways`**

L3 Cache number of ways (sets are implicit from size).

Type: `uint16_t`

Default value: `16`

### **`l3cache-write_access_latency`**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size `l3cache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-write_latency`**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l3cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`memory_tagging_support_level`**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

Type: `uint8_t`

Default value: `3`

### **`mpmm_accumulator_multiplier`**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator.value of `n` means the accumulator will use  $(n * \text{accumulator value})$  to calculate the mpmm threshold (MPMM).is provided as a fast model workaround to handle cases where execution of quantums in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

Type: `uint8_t`

Default value: 1

### **num\_acp**

Number of ACP ports.

Type: uint8\_t

Default value: 0

### **num\_nodes**

Number of transport nodes. Zero implies direct-connect configuration.

Type: uint8\_t

Default value: 1

### **pmu-num\_counters**

Number of PMU counters implemented.

Type: uint8\_t

Default value: 6

### **pseudo\_fault\_generation\_feature\_register**

ARMv8.4 Standard Pseudo-fault generation feature register values. JSON schema for the parameter value is: [{"OF":false, "UC":false, "UEU":false, "UER":false, "UEO":false, "DE":false, "CE":0x0, "CI":false, "ER":false, "PN":false, "AV":false, "MV":false, "SYN":false, "R":false, "NA":false}, other\_pseudo-fault\_generating\_features\_register\_values]. Where OF, UC, UEU, UER, UEO, DE, CI, ER, PN, AV, MV, SYN, and R have valid false(NOT\_SUPPORTED) and true(FEATURE\_CONTROLLABLE), where CE can have 0(NOT\_SUPPORTED), 1(NONSPECIFIC\_CE\_SUPPORTED) and 3(TRANSIENT\_OR\_PERSISTENT\_CE\_SUPPORTED) and NA can have false(component fakes detection on next access) or true(component fakes detection spontaneously). Effective only when ERXFR's INJ field allows it or has\_ras\_fault\_injection is true.

Type: string

Default value: "[{"OF":true, "UC":true, "UEU":false, "UER":false, "UEO":false, "DE":0x1, "CE":0x1, "CI":true, "ER":false, "PN":true, "AV":false, "MV":true, "SYN":true, "R":true}, {"OF":false, "UC":true, "UEU":false, "UER":false, "UEO":false, "DE":0x1, "CE":0x1, "CI":false, "ER":false, "PN":false, "AV":false, "MV":false, "SYN":false, "R":true}]"

### **ptw\_latency**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

**stage12\_tlb\_size**

Number of stage1+2 tlb entries. Valid values are 0 or >= 4.

Type: `uint32_t`

Default value: `0x80`

**tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

**tlbi\_stall\_enabled**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: `bool`

Default value: `false`

**trace\_physical\_registers\_when\_host\_virtualisation\_enabled**

Trace sysreg accesses to physical register accessed (0=disabled, 1=Trace only ELR/SPSR\_EL1 as ELR/SPSR\_EL2, 2=Trace all redirected registers as physical registers affecting all features inc. the host virtualised registers.

Type: `uint8_t`

Default value: `1`

**treat\_PAC\_as\_NOP**

Non-architecture parameter. Treat Pointer Authentication as **NOP**. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are **NOP**. Branch, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

Type: `bool`

Default value: `false`

**walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

## 3.102 ARMCortexX4CT

Defined in `LISA/ARMCortexX4CT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following parameters were added:

- `has_delayed_dbgreg`
- `has_delayed_sysreg`

### About ARMCortexX4CT

The model supports the following features:

- DynamIQ (DSU) system registers.
- DynamIQ r3p0.
- L2 cache is supported at the per-core level only and there is no implementation of Core-Complex with shared L2 cache yet.
- P-Channel for the cluster and for each core.
- Optional peripheral port.
- L3 cache partition.
- Per-core clock.
- `BROADCASTPERSIST` pin.
- Utility bus.
- Support for Interrupt signals from SPE is added as `pmbirq[8]`.

Support for the following features is planned for a future release:

- `BROADCASTCACHEMAINTPOU` pin.
- `COREINSTRRET`, `COREINSTRRUN`, and `nPMBIRQ` signals.
- Each thread has its own L1Cache and L2Cache instead of a common cache that is shared by all threads of the core.
- Per-thread parameters are not supported but signals are implemented.



The `cfgsdisable` signal will be removed in a future release.

The following features will not be implemented:

- DynamIQ features that are irrelevant to the programmers' view simulation, for example:
  - Automatic CPU retention mode.
  - Level-3 Cache RAM retention.
- 256-bit wide output transactions.
- ECC and parity schemes are hardware-specific so are not supported.
- Self-test features (MBIST).
- Latency configuration.
- Snoop filtering.
- Cache stashing capability.
- Embedded Logic Analyzer (ELA).

This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (`crypto.dll` or `crypto.so`) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

## Limitations

Some of the SVE instructions do not raise undefined exceptions when SP is used as a source register.

## Iris and MTI instances for ARMCortexX4CT

This model has the following Iris instances:

Name	Instance type
ARMCortexX4CT	Cluster_ARM_Cortex-X4
ARMCortexX4CT.AMU	PVBusLogger
ARMCortexX4CT.AMU.mapper	PVBusMapper
ARMCortexX4CT.DAP	PVBusLogger
ARMCortexX4CT.DAP.mapper	PVBusMapper
ARMCortexX4CT.DSU	DSU-120
ARMCortexX4CT.DSU.PPU_cluster	PPUv1
ARMCortexX4CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexX4CT.DSU.PPU_core0	PPUv1
ARMCortexX4CT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexX4CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexX4CT.DSU.mpam_busslave	PVBusSlave

Name	Instance type
ARMCortexX4CT.DSU.shared_cache	PVCache
ARMCortexX4CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexX4CT.DSU.shared_cache.upstream[Y] (where Y = 0–4)	PVBusSlave
ARMCortexX4CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexX4CT.MMAP	PVBusLogger
ARMCortexX4CT.MMAP.mapper	PVBusMapper
ARMCortexX4CT.RAS	PVBusLogger
ARMCortexX4CT.RAS.mapper	PVBusMapper
ARMCortexX4CT.cpu0	ARM_Cortex-X4
ARMCortexX4CT.cpu0.UTLB	TLB
ARMCortexX4CT.cpu0.debug_rom	debug_rom
ARMCortexX4CT.cpu0.dtlb	TLB
ARMCortexX4CT.cpu0.l1dcache	PVCache
ARMCortexX4CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexX4CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexX4CT.cpu0.l1licache	PVCache
ARMCortexX4CT.cpu0.l1licache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexX4CT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMCortexX4CT.cpu0.l2cache	PVCache
ARMCortexX4CT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexX4CT.cpu0.l2cache.upstream[U] (where U = 0–1)	PVBusSlave
ARMCortexX4CT.ext_bus	PVBusLogger
ARMCortexX4CT.ext_bus.mapper	PVBusMapper
ARMCortexX4CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexX4CT.global_debug_rom	debug_rom
ARMCortexX4CT.secondary_debug_rom	debug_rom
ARMCortexX4CT.sve	ScalableVectorExtension

This model has the following MTI trace components:

Name	Component type
ARMCortexX4CT.AMU	PVBusLogger
ARMCortexX4CT.AMU.mapper	PVBusMapper
ARMCortexX4CT.DAP	PVBusLogger
ARMCortexX4CT.DAP.mapper	PVBusMapper
ARMCortexX4CT.DSU	DSU-120
ARMCortexX4CT.DSU.PPU_cluster	PPUv1
ARMCortexX4CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexX4CT.DSU.PPU_core0	PPUv1
ARMCortexX4CT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexX4CT.DSU.l3_flusher	AsyncCacheFlushUnit

Name	Component type
ARMCortexX4CT.DSU.mpam_busslave	PVBusSlave
ARMCortexX4CT.DSU.shared_cache	PVCache
ARMCortexX4CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexX4CT.DSU.shared_cache.upstream[Y] (where Y = 0-4)	PVBusSlave
ARMCortexX4CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexX4CT.MMAP	PVBusLogger
ARMCortexX4CT.MMAP.mapper	PVBusMapper
ARMCortexX4CT.RAS	PVBusLogger
ARMCortexX4CT.RAS.mapper	PVBusMapper
ARMCortexX4CT.cpu0	ARM_Cortex-X4
ARMCortexX4CT.cpu0.UTLB	TLB
ARMCortexX4CT.cpu0.l1dcache	PVCache
ARMCortexX4CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexX4CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexX4CT.cpu0.l1icache	PVCache
ARMCortexX4CT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexX4CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexX4CT.cpu0.l2cache	PVCache
ARMCortexX4CT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexX4CT.cpu0.l2cache.upstream[U] (where U = 0-1)	PVBusSlave
ARMCortexX4CT.ext_bus	PVBusLogger
ARMCortexX4CT.ext_bus.mapper	PVBusMapper
ARMCortexX4CT.gic_cpuif_decoder_cluster	GIcV3CPUInterfaceDecoder

## Ports for ARMCortexX4CT

Port	Direction	Protocol	Description
acp_s	slave	PVBus	AXI ACP slave port.
AEND0MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND2MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND3MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART2MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).



Port	Direction	Protocol	Description
broadcastatomic	slave	Signal	CHI defined pins.
broadcastcachemaint	slave	Signal	ACE defined pins.
broadcastouter	slave	Signal	ACE defined pins.
broadcastpersist	slave	Signal	CHI defined pins.
cfgend	slave	Signal	This signal is for EE bit initialisation.
cfgsdisable	slave	Signal	This signal disables write access to some secure Interrupt Controller registers.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcs_m_pchannel	master	PChannel	Cluster PCSM signal
cluster_powerdown_out	master	Signal	DEPRECATED - An output from PPU that informs thermal controller of the cluster power info
cluster_ppu_hw_stat	master	Value	Cluster PPU power state
clusterid	slave	Value	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set ManagerID64 of the core.
clusterpmuirq	master	Signal	DynamiQ pmu irq
CNTHPIRQ	master	Signal	Timer signals to SOC.
CNTHPSIRQ	master	Signal	Timer signals to SOC
CNTHVIRQ	master	Signal	Timer signals to SOC.
CNTHVSIQ	master	Signal	Timer signals to SOC
CNTPNSIRQ	master	Signal	Timer signals to SOC.
CNTPSIQ	master	Signal	Timer signals to SOC.
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.
CNTVIRQ	master	Signal	Timer signals to SOC.
commirq	master	Signal	Interrupt signal from debug communications channel.
core_clk_in	slave	ClockSignal	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
core_pcs_m_pchannel	master	PChannel	Core PCSM signals
core_powerdown_out	master	Signal	DEPRECATED - An output from PPU that informs thermal controller of the core power info
core_ppu_hw_stat	master	Value	Notify the power state change inside the PPU.
coreerrirq	master	Signal	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq	master	Signal	Fault indicator for a detected 1 or 2 bit ECC error

Port	Direction	Protocol	Description
coreinstrrun	master	Signal	Core Running State
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
ctidbgirq	master	Signal	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	slave	Signal	External debug interface.
dbgnopwrdown	master	Signal	No power-down request.
dbgpwrupreq	master	Signal	Debug power up request.
defaultMP	slave	Signal	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main master interface ports.
dev_debug_s	slave	PVBus	External debug interface.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.
gicreset	master	Signal	An output from PPU that can be used to reset GICCLK domain components
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	slave	PVBus	External debug interface.
pmbirq	master	Signal	Interrupt signal from the statistical profiling unit.
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	master	Signal	PPU cluster interrupt.
ppu_cluster_isolate	master	Signal	PPU Cluster Domain isolation control.
ppu_cluster_wakerequest	slave	Signal	PPU cluster wake request signal.
ppu_core_irq	master	Signal	PPU core interrupt.
ppu_core_isolate	master	Signal	PPU Core Domain isolation control.
ppu_core_wakerequest	slave	Signal	PPU core wake request signal.
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	master	PVBus	The core will generate bus requests on this port.
pvbus_periph_m	master	PVBus	The core can generate peripheral bus request on this port.
rei	slave	Signal	Per core RAM Error Interrupt.
reset	slave	Signal	Reset signal to cluster.
rvbaraddr	slave	Value_64	Reset vector base address.
sci_m	master	SystemCoherencyInterface	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei	slave	Signal	Per core System Error physical pins.
spiden	slave	Signal	External debug interface.

Port	Direction	Protocol	Description
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq	master	Signal	Interrupt signal from the trace buffer unit.
utility_bus_s	slave	PVBus	Utility bus slave
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfiq	slave	Signal	Virtualised FIQ.
virq	slave	Signal	Virtualised IRQ.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Per core virtual System Error physical pins.

## Parameters for ARMCortexX4CT

### **cpuX.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **cpuX.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **cpuX.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **cpuX.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **cpuX.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`cpuX.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**cpuX.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**cpuX.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**cpuX.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**cpuX.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**cpuX.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**cpuX.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**cpuX.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**cpuX.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**cpuX.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**cpuX.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**cpuX.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**cpuX.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**cpuX.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**cpuX.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**cpuX.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**cpuX.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: 0xf000000

#### **cpuX.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

#### **cpuX.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

#### **cpuX.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: true

#### **AEND0\_DEFAULT**

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

Type: uint64\_t

Default value: 0x0

#### **AEND1\_DEFAULT**

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

Type: uint64\_t

Default value: 0x0

#### **AEND2\_DEFAULT**

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

Type: uint64\_t



Default value: 0x0

### **AEND3\_DEFAULT**

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

Type: uint64\_t

Default value: 0x0

### **ASTART0\_DEFAULT**

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

Type: uint64\_t

Default value: 0x0

### **ASTART1\_DEFAULT**

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

Type: uint64\_t

Default value: 0x0

### **ASTART2\_DEFAULT**

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

Type: uint64\_t

Default value: 0x0

### **ASTART3\_DEFAULT**

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

Type: uint64\_t

Default value: 0x0

### **BROADCASTATOMIC**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: bool

Default value: `true`

### **BROADCASTCACHEMAINT**

Enable broadcasting of cache maintenance operations to downstream caches. The `broadcastcachemaint` signal will override this value if used.

Type: `bool`

Default value: `false`

### **BROADCASTOUTER**

Enable broadcasting of Outer Shareable transactions. The `broadcastouter` signal will override this value if used.

Type: `bool`

Default value: `false`

### **BROADCASTPERSIST**

Enable broadcasting of cache clean to the point of persistence operations. The `broadcastpersist` signal will override this value if used.

Type: `bool`

Default value: `true`

### **CLUSTER\_ID**

The Cluster ID maps to the affinity levels of the `MPIDR_EL1` register and is evaluated based on the `MPIDR_EL1` layout. If `MPIDR_EL1` supports 16-bit cluster affinity levels, bits [15:8] map to `IDRAFF3`, while bits [7:0] map to `IDRAFF2`. If `MPIDR_EL1` supports 24-bit cluster affinity levels, the bits [23:16] map to `IDRAFF3`, bits [15:8] map to `IDRAFF2`, and bits [7:0] map to `IDRAFF1`. This configuration also updates all relevant component `DEVAFF` registers and is used to set the `ManagerID64` of the core.

Type: `uint32_t`

Default value: `0x0`

### **CMO\_broadcast\_when\_cache\_state\_modelling\_disabled**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. This value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain events even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required.

Type: `uint8_t`

Default value: 1

**CPUCFR**

Value of CPU Configuration Register.

Type: `uint64_t`

Default value: `0x0`

**GICDISABLE**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`

Default value: `true`

**NUM\_CORES**

Number of cores per cluster.

Type: `uint8_t`

Default value: 1

**bus\_type**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

Type: `uint8_t`

Default value: 0

**core\_power\_on\_by\_default**

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPUs which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

Type: `bool`

Default value: `false`

**cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-maintenance\_latency**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-miss\_latency**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-prefetch\_enabled**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

Type: `bool`

Default value: `false`

**dcache-read\_access\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-size**

L1 D-Cache size in bytes.

Type: `uint32_t`

Default value: `0x10000`

**dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-state\_modelled**

Set whether D-cache has stateful implementation.

Type: `bool`

Default value: `false`

**dcache-write\_access\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **default\_opmode**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

Type: `uint8_t`

Default value: 4

### **diagnostics**

Enable DynamIQ diagnostic messages.

Type: `bool`

Default value: `false`

### **ecv\_support\_level**

Implement Enhanced Counter Virtualization feature from ARMv8.6. 0, Not supported. 1, fully supported without CNTPOFF. 2, fully supported with CNTPOFF (FEAT\_ECV).

Type: `uint8_t`

Default value: 2

### **enable\_simulation\_performance\_optimizations**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of `stage12_tlb_size` parameter to 1024).

Type: `bool`

Default value: `true`

### **ete.CLAIMTAGS**

Number of claim tags.

Type: `uint8_t`

Default value: 4

### **ete.MAX\_INST\_PER\_Q**

Maximum limit for the number of instructions implied by a Q element.

Type: `uint16_t`

Default value: 0x1

**ete.NumberOfRSPairs**

Number of resource selector pairs.

Type: `uint8_t`

Default value: 8

**ete.PIDR\_CM0D**

TRCPIDR CM0D value.

Type: `uint8_t`

Default value: 0

**ete.Q\_CADENCE**

Number of instruction blocks traced between two Q elements.

Type: `uint16_t`

Default value: 0x1

**ete.RES0\_STATEFUL**

Whether **RES0** bits are stateful or **RAZ/WI**.

Type: `bool`

Default value: `false`

**ete.RETSTACK**

Return stack depth.

Type: `uint8_t`

Default value: 3

**ete.SIM\_OVERFLOW\_GRANULARITY**

Number of instruction blocks in each granule, for simulated overflow.

Type: `uint32_t`

Default value: 0x64

**ete.SIM\_OVERFLOW\_PERCENTAGE**

Percentage of instruction blocks lost in each granule, for simulated overflow.

Type: `uint8_t`

Default value: 0

**ete.SOURCE\_ADDRESS**

Allow generation of source address elements.

Type: `bool`

Default value: `false`

**ete.TRACE\_OUTPUT**

File to which to write trace byte stream.

Type: `string`

Default value: `N/A`

**ete.TRCSRTA\_FORCED\_EXCEP**

TRCSRTA value for a forcibly traced exception.

Type: `bool`

Default value: `false`

**ext\_abort\_so\_write\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `uint8_t`

Default value: 2

**force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks**

Force MTE tag accesses to **RAZ/WI** and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

Type: `bool`

Default value: `false`

**force\_zero\_PSTATE\_PAN**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`



**force\_zero\_mpam\_partid\_and\_pmg**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

Type: `uint8_t`

Default value: `0`

**has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

**has\_enhanced\_pan**

Implements Armv8.7 Enhanced PAN feature (FEAT\_PAN3) values of this parameter are:- 1, feature is implemented if ARMv8.7 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: `2`

**has\_ete**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (-plugin or -P).

Type: `bool`

Default value: `false`

**has\_peripheral\_port**

If true, an additional AXI peripheral port is configured.

Type: `bool`

Default value: `false`

**has\_statistical\_profiling**

Whether Statistical Based Profiling is implemented (FEAT\_SPE).

Type: `bool`

Default value: `true`

**has\_v8\_7\_spe\_inverted\_filtering**

Where FEAT\_SPEv1p2 is implemented, whether inverted filtering by events is implemented (represented by PMISDR.FnE).

Type: `bool`

Default value: `false`

**has\_v8\_7\_spe\_previous\_branch\_target**

Where FEAT\_SPEv1p2 is implemented, whether the optional branch target feature is implemented (FEAT\_SPE\_PBT).

Type: `bool`

Default value: `false`

**icache-hit\_latency**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

Type: `bool`

Default value: `false`

**icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when icache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**icache-size**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: `0x10000`

**icache-state\_modelled**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

**instruction\_tlb\_size**

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

Type: `uint32_t`

Default value: 0x0

### **invalidate\_code\_cache\_on\_icache\_cmo**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: uint8\_t

Default value: 0

### **l3cache-size**

L3 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

### **l3cache-ways**

L3 Cache number of ways (sets are implicit from size).

Type: uint16\_t

Default value: 16

### **log2\_trace\_buffer\_alignment**

Log2 of trace buffer alignment constraint for output buffer (0->1B ... 11->2Kib).

Type: uint8\_t

Default value: 6

### **memory\_tagging\_support\_level**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

Type: uint8\_t

Default value: 3

### **mpam\_has\_altsp**

DEPRECATED: MPAMIDR\_EL1.HAS\_ALTSP is required when FEAT\_RME is implemented.

Type: bool

Default value: false

**mpamidr\_has\_force\_ns**

Whether MPAMIDR\_EL1.HAS\_FORCE\_NS bit is set or clear.values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.6 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

**mpamidr\_has\_sdeflt**

Whether MPAMIDR\_EL1.HAS\_SDEFLT bit is set or clear.values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.6 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

**mpamidr\_has\_tidr**

Whether MPAMIDR\_EL1.HAS\_TIDR bit is set or clear.values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.6 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

**mpmm\_accumulator\_multiplier**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator.value of n means the accumulator will use (n \* accumulator value) to calculate the mpmm threshold (MPMM).is provided as a fast model workaround to handle cases where execution of quantum in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

Type: uint8\_t

Default value: 1

**num\_acp**

Number of ACP ports.

Type: uint8\_t

Default value: 0

**num\_nodes**

Number of transport nodes. Zero implies direct-connect configuration.

Type: uint8\_t

Default value: 1

**pmu-num\_counters**

Number of PMU counters implemented.

Type: `uint8_t`

Default value: 31

**ptw\_latency**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

**stage12\_tlb\_size**

Number of stage1+2 tlb entries. Valid values are 0 or >= 4.

Type: `uint32_t`

Default value: 0x80

**tcr\_txsz\_undersize\_should\_fault**

If large VA is not supported, Whether undersized TxSZ value should generate translation fault.

Type: `bool`

Default value: `false`

**tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

**tlbi\_stall\_enabled**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: `bool`

Default value: `false`

**treat\_PAC\_as\_NOP**

Non-architecture parameter. Treat Pointer Authentication as **NOP**. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are **NOP**. Branch, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

Type: `bool`

Default value: `false`

**walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

## 3.103 ARMCortexX925CT

Defined in `LISA/ARMCortexX925CT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following parameters were added:

- `has_delayed_dbgreg`
- `has_delayed_sysreg`

### About ARMCortexX925CT

A DSU-120 DynamIQ cluster containing a configurable number of Cortex-X925 cores.

The number of cores in the cluster is configurable using the following parameter:

**NUM\_CORES**

Possible values are 1-14

The following DSU/CPU features will not be implemented:

- DynamIQ features that are irrelevant to the programmers' view simulation.
- 256-bit wide output transactions.

- ECC and parity schemes are hardware-specific so are not supported.
- Self-test features (MBIST).
- Snoop filtering.
- Cache stashing capability.
- Embedded Logic Analyzer (ELA).

This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (`crypto.dll` or `crypto.so`) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

## Iris and MTI instances for ARM CortexX925CT

This model has the following Iris instances:

Name	Instance type
ARM CortexX925CT	Cluster_ARM_Cortex-X925
ARM CortexX925CT.AMU	PVBusLogger
ARM CortexX925CT.AMU.mapper	PVBusMapper
ARM CortexX925CT.DAP	PVBusLogger
ARM CortexX925CT.DAP.mapper	PVBusMapper
ARM CortexX925CT.DSU	DSU-120
ARM CortexX925CT.DSU.PPU_cluster	PPUv1
ARM CortexX925CT.DSU.PPU_cluster.busslave	PVBusSlave
ARM CortexX925CT.DSU.PPU_core0	PPUv1
ARM CortexX925CT.DSU.PPU_core0.busslave	PVBusSlave
ARM CortexX925CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARM CortexX925CT.DSU.mpam_busslave	PVBusSlave
ARM CortexX925CT.DSU.shared_cache	PVCache
ARM CortexX925CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARM CortexX925CT.DSU.shared_cache.upstream[Y] (where Y = 0-4)	PVBusSlave
ARM CortexX925CT.DSU.utility_slave[0]	PVBusSlave
ARM CortexX925CT.MMAP	PVBusLogger
ARM CortexX925CT.MMAP.mapper	PVBusMapper
ARM CortexX925CT.RAS	PVBusLogger
ARM CortexX925CT.RAS.mapper	PVBusMapper
ARM CortexX925CT.cpu0	ARM_Cortex-X925
ARM CortexX925CT.cpu0.UTLB	TLB
ARM CortexX925CT.cpu0.debug_rom	debug_rom
ARM CortexX925CT.cpu0.dtlb	TLB
ARM CortexX925CT.cpu0.l1dcache	PVCache
ARM CortexX925CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARM CortexX925CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARM CortexX925CT.cpu0.l1icache	PVCache



Name	Instance type
ARMCortexX925CT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexX925CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexX925CT.cpu0.l2cache	PVCache
ARMCortexX925CT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexX925CT.cpu0.l2cache.upstream[U] (where $U = 0-1$ )	PVBusSlave
ARMCortexX925CT.ext_bus	PVBusLogger
ARMCortexX925CT.ext_bus.mapper	PVBusMapper
ARMCortexX925CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMCortexX925CT.global_debug_rom	debug_rom
ARMCortexX925CT.secondary_debug_rom	debug_rom
ARMCortexX925CT.sve	ScalableVectorExtension

This model has the following MTI trace components:

Name	Component type
ARMCortexX925CT.AMU	PVBusLogger
ARMCortexX925CT.AMU.mapper	PVBusMapper
ARMCortexX925CT.DAP	PVBusLogger
ARMCortexX925CT.DAP.mapper	PVBusMapper
ARMCortexX925CT.DSU	DSU-120
ARMCortexX925CT.DSU.PPU_cluster	PPUv1
ARMCortexX925CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMCortexX925CT.DSU.PPU_core0	PPUv1
ARMCortexX925CT.DSU.PPU_core0.busslave	PVBusSlave
ARMCortexX925CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMCortexX925CT.DSU.mpam_busslave	PVBusSlave
ARMCortexX925CT.DSU.shared_cache	PVCache
ARMCortexX925CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexX925CT.DSU.shared_cache.upstream[Y] (where $Y = 0-4$ )	PVBusSlave
ARMCortexX925CT.DSU.utility_slave[0]	PVBusSlave
ARMCortexX925CT.MMAP	PVBusLogger
ARMCortexX925CT.MMAP.mapper	PVBusMapper
ARMCortexX925CT.RAS	PVBusLogger
ARMCortexX925CT.RAS.mapper	PVBusMapper
ARMCortexX925CT.cpu0	ARM_Cortex-X925
ARMCortexX925CT.cpu0.UTLB	TLB
ARMCortexX925CT.cpu0.l1dcache	PVCache
ARMCortexX925CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexX925CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMCortexX925CT.cpu0.l1icache	PVCache
ARMCortexX925CT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster

Name	Component type
ARMCortexX925CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMCortexX925CT.cpu0.l2cache	PVCache
ARMCortexX925CT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMCortexX925CT.cpu0.l2cache.upstream[U] (where U = 0-1)	PVBusSlave
ARMCortexX925CT.ext_bus	PVBusLogger
ARMCortexX925CT.ext_bus.mapper	PVBusMapper
ARMCortexX925CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

### Ports for ARMCortexX925CT

Port	Direction	Protocol	Description
acp_s	slave	PVBus	AXI ACP slave port.
AEND0MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND2MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND3MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART2MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	slave	Signal	CHI defined pins.
broadcastcachemaint	slave	Signal	ACE defined pins.
broadcastouter	slave	Signal	ACE defined pins.
broadcastpersist	slave	Signal	CHI defined pins.
cfgend	slave	Signal	This signal is for EE bit initialisation.
cfgsdisable	slave	Signal	This signal disables write access to some secure Interrupt Controller registers.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcsn_pchannel	master	PChannel	Cluster PCSM signal
cluster_powerdown_out	master	Signal	DEPRECATED - An output from PPUs that informs thermal controller of the cluster power info
cluster_ppu_hw_stat	master	Value	Cluster PPU power state

Port	Direction	Protocol	Description
clusterid	slave	Value	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set ManagerID64 of the core.
clusterpmuirq	master	Signal	DynamiQ pmu irq
CNTHPIRQ	master	Signal	Timer signals to SOC.
CNTHPSIRQ	master	Signal	Timer signals to SOC
CNTHVIRQ	master	Signal	Timer signals to SOC.
CNTHVSIRQ	master	Signal	Timer signals to SOC
CNTPNSIRQ	master	Signal	Timer signals to SOC.
CNTPSIRQ	master	Signal	Timer signals to SOC.
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.
CNTVIRQ	master	Signal	Timer signals to SOC.
commirq	master	Signal	Interrupt signal from debug communications channel.
core_clk_in	slave	ClockSignal	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
core_pcsn_pchannel	master	PChannel	Core PCSM signals
core_powerdown_out	master	Signal	DEPRECATED - An output from PPUs that informs thermal controller of the core power info
core_ppu_hw_stat	master	Value	Notify the power state change inside the PPU.
coreerrirq	master	Signal	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq	master	Signal	Fault indicator for a detected 1 or 2 bit ECC error
coreinstrrun	master	Signal	Core Running State
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
ctidbgirq	master	Signal	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	slave	Signal	External debug interface.
dbgnopwrdown	master	Signal	No power-down request.
dbgprupreq	master	Signal	Debug power up request.
defaultMP	slave	Signal	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main master interface ports.
dev_debug_s	slave	PVBus	External debug interface.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.

Port	Direction	Protocol	Description
gicreset	master	Signal	An output from PPU that can be used to reset GICCLK domain components
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	slave	PVBus	External debug interface.
pmbirq	master	Signal	Interrupt signal from the statistical profiling unit.
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	master	Signal	PPU cluster interrupt.
ppu_cluster_isolate	master	Signal	PPU Cluster Domain isolation control.
ppu_cluster_wakerequest	slave	Signal	PPU cluster wake request signal.
ppu_core_irq	master	Signal	PPU core interrupt.
ppu_core_isolate	master	Signal	PPU Core Domain isolation control.
ppu_core_wakerequest	slave	Signal	PPU core wake request signal.
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	master	PVBus	The core will generate bus requests on this port.
pvbus_periph_m	master	PVBus	The core can generate peripheral bus request on this port.
rei	slave	Signal	Per core RAM Error Interrupt.
reset	slave	Signal	Reset signal to cluster.
rvbaraddr	slave	Value_64	Reset vector base address.
sci_m	master	SystemCoherencyInterface	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei	slave	Signal	Per core System Error physical pins.
spiden	slave	Signal	External debug interface.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq	master	Signal	Interrupt signal from the trace buffer unit.
utility_bus_s	slave	PVBus	Utility bus slave
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfiq	slave	Signal	Virtualised FIQ.
virq	slave	Signal	Virtualised IRQ.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Per core virtual System Error physical pins.

## Parameters for ARMCortexX925CT

### **cpuX.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **cpuX.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **cpuX.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **cpuX.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **cpuX.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **cpuX.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**cpuX.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**cpuX.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**cpuX.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**cpuX.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**cpuX.l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x80000`

**cpuX.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**cpuX.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

**cpuX.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**cpuX.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**cpuX.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**cpuX.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

**cpuX.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**cpuX.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**cpuX.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**cpuX.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**cpuX.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`



Default value: N/A

**cpuX.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: true

**cpuX.semihosting-heap\_base**

Virtual address of heap base.

Type: uint64\_t

Default value: 0x0

**cpuX.semihosting-heap\_limit**

Virtual address of top of heap.

Type: uint64\_t

Default value: 0xf000000

**cpuX.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: uint64\_t

Default value: 0x10000000

**cpuX.semihosting-stack\_limit**

Virtual address of stack limit.

Type: uint64\_t

Default value: 0xf000000

**cpuX.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**cpuX.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**cpuX.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**AEND0\_DEFAULT**

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

Type: `uint64_t`

Default value: `0x0`

**AEND1\_DEFAULT**

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

Type: `uint64_t`

Default value: `0x0`

**AEND2\_DEFAULT**

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

Type: `uint64_t`

Default value: `0x0`

**AEND3\_DEFAULT**

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART0\_DEFAULT**

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART1\_DEFAULT**

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART2\_DEFAULT**

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART3\_DEFAULT**

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

Type: `uint64_t`

Default value: `0x0`

**BROADCASTATOMIC**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`

Default value: `true`

**BROADCASTCACHEMAINT**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

Type: `bool`

Default value: `false`

**BROADCASTOUTER**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

Type: `bool`

Default value: `false`

**BROADCASTPERSIST**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: `bool`

Default value: `true`

**CLUSTER\_ID**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the ManagerID64 of the core.

Type: `uint32_t`

Default value: `0x0`

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. This value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required.

Type: `uint8_t`

Default value: `1`

**CPUCFR**

Value of CPU Configuration Register.

Type: `uint64_t`

Default value: `0x0`

**GICDISABLE**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`

Default value: `true`

**NUM\_CORES**

Number of cores per cluster.

Type: `uint8_t`

Default value: 1

**bus\_type**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

Type: `uint8_t`

Default value: 0

**core\_power\_on\_by\_default**

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPUs which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

Type: `bool`

Default value: `false`

**cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-maintenance\_latency**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-miss\_latency**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-prefetch\_enabled**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

Type: `bool`

Default value: `false`

**dcache-read\_access\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **dcache-size**

L1 D-Cache size in bytes.

Type: `uint32_t`

Default value: `0x10000`

### **dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **dcache-state\_modelled**

Set whether D-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **dcache-write\_access\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **default\_opmode**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

Type: `uint8_t`

Default value: 4

### **diagnostics**

Enable DynamIQ diagnostic messages.

Type: `bool`

Default value: `false`

### **ecv\_support\_level**

Implement Enhanced Counter Virtualization feature from ARMv8.6. 0, Not supported. 1, fully supported without CNTPOFF. 2, fully supported with CNTPOFF (FEAT\_ECV).

Type: `uint8_t`

Default value: 2

### **enable\_simulation\_performance\_optimizations**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of `stage12_tlb_size` parameter to 1024).

Type: `bool`

Default value: `true`

### **ete.CLAIMTAGS**

Number of claim tags.

Type: `uint8_t`

Default value: 4

### **ete.MAX\_INST\_PER\_Q**

Maximum limit for the number of instructions implied by a Q element.

Type: `uint16_t`

Default value: `0x1`

### **ete.NumberOfRSPairs**

Number of resource selector pairs.

Type: `uint8_t`



Default value: 8

**ete.PIDR\_CM0D**

TRCPIDR CM0D value.

Type: `uint8_t`

Default value: 0

**ete.Q\_CADENCE**

Number of instruction blocks traced between two Q elements.

Type: `uint16_t`

Default value: 0x1

**ete.RES0\_STATEFUL**

Whether **RES0** bits are stateful or **RAZ/WI**.

Type: `bool`

Default value: `false`

**ete.RETSTACK**

Return stack depth.

Type: `uint8_t`

Default value: 3

**ete.SIM\_OVERFLOW\_GRANULARITY**

Number of instruction blocks in each granule, for simulated overflow.

Type: `uint32_t`

Default value: 0x64

**ete.SIM\_OVERFLOW\_PERCENTAGE**

Percentage of instruction blocks lost in each granule, for simulated overflow.

Type: `uint8_t`

Default value: 0

**ete.SOURCE\_ADDRESS**

Allow generation of source address elements.

Type: `bool`

Default value: `false`

### **`ete.TRACE_OUTPUT`**

File to which to write trace byte stream.

Type: `string`

Default value: `N/A`

### **`ete.TRCSRSTA_FORCED_EXCEP`**

TRCSRSTA value for a forcibly traced exception.

Type: `bool`

Default value: `false`

### **`ext_abort_so_write_ras_type`**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `uint8_t`

Default value: 2

### **`force_mte_tag_access_razwi_and_ignore_tag_checks`**

Force MTE tag accesses to **RAZ/WI** and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

Type: `bool`

Default value: `false`

### **`force_zero_PSTATE_PAN`**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

### **`force_zero_mpam_partid_and_pmg`**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

Type: `uint8_t`

Default value: 0

**has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

**has\_enhanced\_pan**

Implements Armv8.7 Enhanced PAN feature (FEAT\_PAN3) values of this parameter are:- 1, feature is implemented if ARMv8.7 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 2

**has\_ete**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (`-plugin` or `-P`).

Type: `bool`

Default value: `false`

**has\_mt\_pmu\_disable\_feature**

Implements Multi-threading PMU disable extension from ARMv8.6 (FEAT\_MTPMU). 0: FEAT\_MTPMU is disabled, 1: FEAT\_MTPMU is enabled if ARMv8.6 is implemented, 2: FEAT\_MTPMU is cherry-picked, 0xF: The feature is disabled and is represented by value 0xF in ID\_AA64DFR0\_EL1.MTPMU.

Type: `uint8_t`

Default value: 0

**has\_peripheral\_port**

If true, an additional AXI peripheral port is configured.

Type: `bool`

Default value: `false`

**has\_statistical\_profiling**

Whether Statistical Based Profiling is implemented (FEAT\_SPE).

Type: `bool`

Default value: `true`

**has\_v8\_7\_spe\_inverted\_filtering**

Where FEAT\_SPEv1p2 is implemented, whether inverted filtering by events is implemented (represented by PMISDR.FnE).

Type: `bool`

Default value: `false`

**has\_v8\_7\_spe\_previous\_branch\_target**

Where FEAT\_SPEv1p2 is implemented, whether the optional branch target feature is implemented (FEAT\_SPE\_PBT).

Type: `bool`

Default value: `false`

**icache-hit\_latency**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

Type: `bool`

Default value: `false`

**icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when icache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**icache-size**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: `0x10000`

**icache-state\_modelled**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

**instruction\_tlb\_size**

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

Type: `uint32_t`

Default value: 0x0

### **invalidate\_code\_cache\_on\_icache\_cmo**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: uint8\_t

Default value: 0

### **l3cache-size**

L3 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

### **l3cache-ways**

L3 Cache number of ways (sets are implicit from size).

Type: uint16\_t

Default value: 16

### **log2\_trace\_buffer\_alignment**

Log2 of trace buffer alignment constraint for output buffer (0->1B ... 11->2Kib).

Type: uint8\_t

Default value: 6

### **memory\_tagging\_support\_level**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

Type: uint8\_t

Default value: 3

### **mpam\_has\_altsp**

DEPRECATED: MPAMIDR\_EL1.HAS\_ALTSP is required when FEAT\_RME is implemented.

Type: bool

Default value: false

**mpamidr\_has\_force\_ns**

Whether MPAMIDR\_EL1.HAS\_FORCE\_NS bit is set or clear.values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.6 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 0

**mpamidr\_has\_sdeflt**

Whether MPAMIDR\_EL1.HAS\_SDEFLT bit is set or clear.values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.6 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

**mpamidr\_has\_tidr**

Whether MPAMIDR\_EL1.HAS\_TIDR bit is set or clear.values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.6 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

**mpmm\_accumulator\_multiplier**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator.value of n means the accumulator will use (n \* accumulator value) to calculate the mpmm threshold (MPMM).is provided as a fast model workaround to handle cases where execution of quantum in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

Type: uint8\_t

Default value: 1

**num\_acp**

Number of ACP ports.

Type: uint8\_t

Default value: 0

**num\_nodes**

Number of transport nodes. Zero implies direct-connect configuration.

Type: uint8\_t

Default value: 1

**pmu-num\_counters**

Number of PMU counters implemented.

Type: `uint8_t`

Default value: 31

**ptw\_latency**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

**stage12\_tlb\_size**

Number of stage1+2 tlb entries. Valid values are 0 or >= 4.

Type: `uint32_t`

Default value: 0x80

**tcr\_txsz\_undersize\_should\_fault**

If large VA is not supported, Whether undersized TxSZ value should generate translation fault.

Type: `bool`

Default value: `false`

**tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

**tlbi\_stall\_enabled**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: `bool`

Default value: `false`



**treat\_PAC\_as\_NOP**

Non-architecture parameter. Treat Pointer Authentication as **NOP**. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are **NOP**. Branch, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

Type: `bool`

Default value: `false`

**walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

## 3.104 ARMNeoverseE1CT

Defined in `LISA/ARMNeoverseE1CT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following parameters were added:

- `has_delayed_dbgreg`
- `has_delayed_sysreg`

### About ARMNeoverseE1CT

The model supports the following features:

- DynamIQ (DSU) system registers.
- Per-core L2 cache.
- A P-Channel for the cluster and for each core.
- Optional peripheral port.
- L3 cache partition.
- Per-core clock.
- `BROADCASTPERSIST` pin.

The model does not support the following features:

- `BROADCASTCACHEMAINTPOU` pin.
- `COREINSTRRET`, `COREINSTRRUN`, or `nPMBIRQ` signals.
- DynamIQ features that are irrelevant to the programmers' view simulation, for example:
  - Automatic CPU retention mode.
  - Level-3 Cache RAM retention.
- 256-bit wide output transactions.
- ECC and parity schemes are hardware-specific so are not supported.
- Self-test features (MBIST).
- Latency configuration.
- Snoop filtering.
- Debug power management signals `DBGPWRDUP` and `DBGIRSTREQ` are not implemented, but `DBGPWRUPREQ` and `DBGNOPWRDWN` are implemented.
- Cache stashing capability.
- Each thread has its own L1Cache and L2Cache instead of a common cache that is shared by all threads of the core.
- Per-thread parameters are not yet supported but signals are implemented.

This model has a variable number of cores per cluster, specified using the `NUM_CORES` parameter.

The per-core parameters are preceded by `cpun.`, where `n` identifies the core (0-3).

This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (`crypto.dll` or `crypto.so`) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

## Iris and MTI instances for ARMNeoverseE1CT

This model has the following Iris instances:

Name	Instance type
ARMNeoverseE1CT	Cluster_ARM_Neoverse-E1
ARMNeoverseE1CT.AMU	PVBusLogger
ARMNeoverseE1CT.AMU.mapper	PVBusMapper
ARMNeoverseE1CT.DAP	PVBusLogger
ARMNeoverseE1CT.DAP.mapper	PVBusMapper
ARMNeoverseE1CT.DSU	DSU
ARMNeoverseE1CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMNeoverseE1CT.DSU.mpam_busslave	PVBusSlave
ARMNeoverseE1CT.DSU.shared_cache	PVCache
ARMNeoverseE1CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseE1CT.DSU.shared_cache.upstream[Y] (where Y = 0-3)	PVBusSlave

Name	Instance type
ARMNeoverseE1CT.MMAP	PVBusLogger
ARMNeoverseE1CT.MMAP.mapper	PVBusMapper
ARMNeoverseE1CT.RAS	PVBusLogger
ARMNeoverseE1CT.RAS.mapper	PVBusMapper
ARMNeoverseE1CT.cpu0.dtlb	TLB
ARMNeoverseE1CT.cpu0.threadZ (where Z = 0-1)	ARM_Neoverse-E1
ARMNeoverseE1CT.cpu0.threadZ.UTLB (where Z = 0-1)	TLB
ARMNeoverseE1CT.cpu0.thread0.l1dcache	PVCache
ARMNeoverseE1CT.cpu0.thread0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseE1CT.cpu0.thread0.l1dcache.upstream[0]	PVBusSlave
ARMNeoverseE1CT.cpu0.thread0.l1icache	PVCache
ARMNeoverseE1CT.cpu0.thread0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseE1CT.cpu0.thread0.l1icache.upstream[0]	PVBusSlave
ARMNeoverseE1CT.cpu0.thread0.l2cache	PVCache
ARMNeoverseE1CT.cpu0.thread0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseE1CT.cpu0.thread0.l2cache.upstream[V] (where V = 0-1)	PVBusSlave
ARMNeoverseE1CT.ext_bus	PVBusLogger
ARMNeoverseE1CT.ext_bus.mapper	PVBusMapper
ARMNeoverseE1CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMNeoverseE1CT.global_debug_rom	debug_rom

This model has the following MTI trace components:

Name	Component type
ARMNeoverseE1CT.AMU	PVBusLogger
ARMNeoverseE1CT.AMU.mapper	PVBusMapper
ARMNeoverseE1CT.DAP	PVBusLogger
ARMNeoverseE1CT.DAP.mapper	PVBusMapper
ARMNeoverseE1CT.DSU	DSU
ARMNeoverseE1CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMNeoverseE1CT.DSU.mpam_busslave	PVBusSlave
ARMNeoverseE1CT.DSU.shared_cache	PVCache
ARMNeoverseE1CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseE1CT.DSU.shared_cache.upstream[Y] (where Y = 0-3)	PVBusSlave
ARMNeoverseE1CT.MMAP	PVBusLogger
ARMNeoverseE1CT.MMAP.mapper	PVBusMapper
ARMNeoverseE1CT.RAS	PVBusLogger
ARMNeoverseE1CT.RAS.mapper	PVBusMapper
ARMNeoverseE1CT.cpu0.threadZ (where Z = 0-1)	ARM_Neoverse-E1
ARMNeoverseE1CT.cpu0.threadZ.UTLB (where Z = 0-1)	TLB
ARMNeoverseE1CT.cpu0.thread0.l1dcache	PVCache

Name	Component type
ARMNeoverseE1CT.cpu0.thread0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseE1CT.cpu0.thread0.l1dcache.upstream[0]	PVBusSlave
ARMNeoverseE1CT.cpu0.thread0.l1licache	PVCache
ARMNeoverseE1CT.cpu0.thread0.l1licache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseE1CT.cpu0.thread0.l1licache.upstream[0]	PVBusSlave
ARMNeoverseE1CT.cpu0.thread0.l2cache	PVCache
ARMNeoverseE1CT.cpu0.thread0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseE1CT.cpu0.thread0.l2cache.upstream[V] (where V = 0–1)	PVBusSlave
ARMNeoverseE1CT.ext_bus	PVBusLogger
ARMNeoverseE1CT.ext_bus.mapper	PVBusMapper
ARMNeoverseE1CT.gic_cpuif_decoder_cluster	GLCv3CPUInterfaceDecoder

### Ports for ARMNeoverseE1CT

Port	Direction	Protocol	Description
acp_s	slave	PVBus	AXI ACP slave port
AENDMP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
ASTARTMP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	slave	Signal	CHI defined pins.
broadcastcachemaint	slave	Signal	ACE defined pins.
broadcastouter	slave	Signal	ACE defined pins.
broadcastpersist	slave	Signal	CHI defined pins.
cfgend	slave	Signal	This signal if for EE bit initialisation.
cfgsdisable	slave	Signal	This signal disables write access to some secure Interrupt Controller registers.
cfgte	slave	Signal	This signal provides default exception handling state.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
clusterid	slave	Value	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set ManagerID64 of the core.
clusterpmuirq	master	Signal	DynamiQ pmu irq
CNTHPIRQ	master	Signal	Timer signals to SOC.
CNTHVIRQ	master	Signal	Timer signals to SOC.
CNTPNSIRQ	master	Signal	Timer signals to SOC.

Port	Direction	Protocol	Description
CNTPSIRQ	master	Signal	Timer signals to SOC.
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.
CNTVIRQ	master	Signal	Timer signals to SOC.
commirq	master	Signal	Interrupt signal from debug communications channel.
core_clk_in	slave	ClockSignal	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
cp15sdisable	slave	Signal	This signal disables write access to some system control processor registers.
cpuporeset	slave	Signal	Power on reset. Initializes all the processor logic, including debug logic.
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
ctidbgirq	master	Signal	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	slave	Signal	External debug interface.
dbgnopwrdown	master	Signal	These signals relate to core power down.
dbgppwrupreq	master	Signal	Debug power up request.
dev_debug_s	slave	PVBus	External debug interface.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port per thread.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	slave	PVBus	External debug interface.
niden	slave	Signal	External debug interface.
pchannel_cluster	slave	PChannel	PChannel for cluster.
pchannel_core	slave	PChannel	PChannels for cores
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	master	PVBus	The core will generate bus requests on this port.
pvbus_periph_m	master	PVBus	The core can generate peripheral bus request on this port.
rei	slave	Signal	Per core RAM Error Interrupt.
reset	slave	Signal	Raising this signal will put the core into reset mode.
rvbaraddr	slave	Value_64	Reset vector base address.
sci_m	master	SystemCoherencyInterface	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei	slave	Signal	Per core virtual System Error physical pins.
spiden	slave	Signal	External debug interface.
spniden	slave	Signal	External debug interface.

Port	Direction	Protocol	Description
sporeset	slave	Signal	A single cluster-wide power on reset signal for all resettable registers in DynamIQ.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfiq	slave	Signal	Virtualised FIQ.
vinithi	slave	Signal	This signal controls of the location of the exception vectors at reset.
virq	slave	Signal	Virtualised IRQ.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Per core virtual System Error physical pins.

## Parameters for ARMNeoverseE1CT

### **cpuX.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **cpuX.enable\_single\_thread\_at\_reset**

Enable single thread after reset and keep other thread in reset.

Type: `bool`

Default value: `false`

### **cpuX.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **cpuX.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**cpuX.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**cpuX.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**cpuX.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**cpuX.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**cpuX.l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x40000`

**cpuX.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**cpuX.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**cpuX.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**cpuX.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t



Default value: 0xf000

**cpuX.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**cpuX.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**cpuX.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**cpuX.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**cpuX.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**cpuX.semihosting-cwd**

Base directory for semihosting file access.

Type: string

Default value: N/A

**cpuX.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: `true`

**cpuX.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**cpuX.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**cpuX.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**cpuX.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**cpuX.thread0.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian.

Type: `bool`

Default value: `false`

**cpuX.thread0.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

**cpuX.thread0.MPIDR-override**

Override MPIDR value. A nonzero value will override the MT,cluster,cpu,thread ID bits in MPIDR.

Type: `uint64_t`

Default value: 0x0

**cpuX.thread0.RVBARADDR**

Value of RVBAR\_ELx register.

Type: uint64\_t

Default value: 0x0

**cpuX.thread0.VINITHI**

Reset value of SCTLR.V.

Type: bool

Default value: false

**cpuX.thread1.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian.

Type: bool

Default value: false

**cpuX.thread1.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: bool

Default value: false

**cpuX.thread1.MPIDR-override**

Override MPIDR value. A nonzero value will override the MT,cluster,cpu,thread ID bits in MPIDR.

Type: uint64\_t

Default value: 0x0

**cpuX.thread1.RVBARADDR**

Value of RVBAR\_ELx register.

Type: uint64\_t

Default value: 0x0

**cpuX.thread1.VINITHI**

Reset value of SCTLR.V.

Type: bool

Default value: `false`

#### **`cpuX.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

#### **`cpuX.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

#### **`cpuX.vfp-present`**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

#### **`BROADCASTATOMIC`**

Enable broadcasting of atomic operation. The `broadcastatomic` signal will override this value if used.

Type: `bool`

Default value: `true`

#### **`BROADCASTCACHEMAINT`**

Enable broadcasting of cache maintenance operations to downstream caches. The `broadcastcachemaint` signal will override this value if used.

Type: `bool`

Default value: `true`

#### **`BROADCASTOUTER`**

Enable broadcasting of Outer Shareable transactions. The `broadcastouter` signal will override this value if used.

Type: `bool`

Default value: `true`

### **BROADCASTPERSIST**

Enable broadcasting of cache clean to the point of persistence operations. The `broadcastpersist` signal will override this value if used.

Type: `bool`

Default value: `true`

### **CLUSTER\_ID**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the ManagerID64 of the core.

Type: `uint32_t`

Default value: `0x0`

### **CMO\_broadcast\_when\_cache\_state\_modelling\_disabled**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. This value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain even when cache state modelling is disabled. Possible values are:- 0 = All CMOs are broadcast if architecturally required- 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required.

Type: `uint8_t`

Default value: `1`

### **GICDISABLE**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`

Default value: `true`

### **NUM\_CORES**

Number of cores per cluster.

Type: `uint8_t`

Default value: `1`

**cluster\_patch\_level**

Cosmetic change to patch number in CLUSTERIDR. Corresponds to the Y in rXpY.

Type: uint8\_t

Default value: 0

**cluster\_revision\_number**

Cosmetic change to revision number in CLUSTERIDR, Corresponds to the X in rXpY.

Type: uint8\_t

Default value: 0

**cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: uint32\_t

Default value: 0x1

**cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: uint32\_t

Default value: 0x1

**dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**dcache-maintenance\_latency**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**dcache-miss\_latency**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

### **`dcache-prefetch_enabled`**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

Type: `bool`

Default value: `false`

### **`dcache-read_access_latency`**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`dcache-read_latency`**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`dcache-size`**

L1 D-Cache size in bytes.

Type: `uint32_t`

Default value: `0x10000`

### **`dcache-snoop_data_transfer_latency`**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-state\_modelled**

Set whether D-cache has stateful implementation.

Type: `bool`

Default value: `false`

**dcache-write\_access\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**default\_opmode**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

Type: `uint8_t`

Default value: 4

**diagnostics**

Enable DynamIQ diagnostic messages.

Type: `bool`

Default value: `false`

**enable\_simulation\_performance\_optimizations**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of `stage12_tlb_size` parameter to 1024).

Type: `bool`



Default value: `true`

### **`force_zero_PSTATE_PAN`**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

### **`has_acp`**

If true, Accelerator Coherency Port is configured.

Type: `bool`

Default value: `false`

### **`has_delayed_dbgreg`**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

### **`has_delayed_sysreg`**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

### **`has_dot_product`**

Possible values of this parameter are:- 1, feature not implemented.- 2, feature is implemented.

Type: `uint8_t`

Default value: 2

### **`has_peripheral_port`**

If true, additional AXI peripheral port is configured.

Type: `bool`

Default value: `false`

**icache-hit\_latency**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

**icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to

the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`icache-size`**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: `0x10000`

### **`icache-state_modelled`**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **`invalidate_code_cache_on_icache_cmo`**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: `uint8_t`

Default value: `0`

### **`l3cache-hit_latency`**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-maintenance_latency`**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-miss_latency`**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-read_access_latency`**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size `l3cache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-read_latency`**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l3cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-size`**

L3 Cache size in bytes.

Type: `uint32_t`

Default value: `0x400000`

### **`l3cache-snoop_data_transfer_latency`**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-snoop_issue_latency`**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-write\_access\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size l3cache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**l3cache-write\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. l3cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**pchannel\_treat\_simreset\_as\_poreset**

Register core as ON state to cluster with simulation reset.

Type: bool

Default value: false

**periph\_address\_end**

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

Type: uint64\_t

Default value: 0x0

**periph\_address\_start**

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

Type: uint64\_t

Default value: 0x0

**ptw\_latency**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

**tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

**tlbi\_stall\_enabled**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: `bool`

Default value: `false`

**walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

## 3.105 ARMNeoverseN1CT

Defined in `LISA/ARMNeoverseN1CT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r4p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

**Changes in 11.31.15**

The following parameters were added:

- `has_delayed_dbgreg`
- `has_delayed_sysreg`

**About ARMNeoverseN1CT**

The model supports the following features:

- DynamIQ (DSU) system registers.
- Per-core L2 cache.
- A P-Channel for the cluster and for each core.

- Optional peripheral port.
- L3 cache partition.
- Per-core clock.
- BROADCASTPERSIST pin.

The model does not support the following features:

- BROADCASTCACHEMAINTPOU pin.
- COREINSTRRET, COREINSTRRUN, or nPMBIRQ signals.
- DynamIQ features that are irrelevant to the programmers' view simulation, for example:
  - Automatic CPU retention mode.
  - Level-3 Cache RAM retention.
- 256-bit wide output transactions.
- ECC and parity schemes are hardware-specific so are not supported.
- Self-test features (MBIST).
- Latency configuration.
- Snoop filtering.
- Debug power management signals DBGWRUPREQ and DBGWRUPREQ are not implemented, but DBGWRUPREQ and DBGWRUPREQ are implemented.
- Cache stashing capability.

This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (crypto.dll or crypto.so) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

## Iris and MTI instances for ARMNeoverseN1CT

This model has the following Iris instances:

Name	Instance type
ARMNeoverseN1CT	Cluster_ARM_Neoverse-N1
ARMNeoverseN1CT.AMU	PVBusLogger
ARMNeoverseN1CT.AMU.mapper	PVBusMapper
ARMNeoverseN1CT.DAP	PVBusLogger
ARMNeoverseN1CT.DAP.mapper	PVBusMapper
ARMNeoverseN1CT.DSU	DSU
ARMNeoverseN1CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMNeoverseN1CT.DSU.mpam_busslave	PVBusSlave
ARMNeoverseN1CT.DSU.shared_cache	PVCache
ARMNeoverseN1CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseN1CT.DSU.shared_cache.upstream[Y] (where Y = 0-3)	PVBusSlave
ARMNeoverseN1CT.MMAP	PVBusLogger

Name	Instance type
ARMNeoverseN1CT.MMAP.mapper	PVBusMapper
ARMNeoverseN1CT.RAS	PVBusLogger
ARMNeoverseN1CT.RAS.mapper	PVBusMapper
ARMNeoverseN1CT.cpu0	ARM_Neoverse-N1
ARMNeoverseN1CT.cpu0.UTLB	TLB
ARMNeoverseN1CT.cpu0.debug_rom	debug_rom
ARMNeoverseN1CT.cpu0.dtlb	TLB
ARMNeoverseN1CT.cpu0.l1dcache	PVCache
ARMNeoverseN1CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseN1CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMNeoverseN1CT.cpu0.l1icache	PVCache
ARMNeoverseN1CT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseN1CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMNeoverseN1CT.cpu0.l2cache	PVCache
ARMNeoverseN1CT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseN1CT.cpu0.l2cache.upstream[U] (where U = 0-1)	PVBusSlave
ARMNeoverseN1CT.ext_bus	PVBusLogger
ARMNeoverseN1CT.ext_bus.mapper	PVBusMapper
ARMNeoverseN1CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMNeoverseN1CT.global_debug_rom	debug_rom
ARMNeoverseN1CT.secondary_debug_rom	debug_rom

This model has the following MTI trace components:

Name	Component type
ARMNeoverseN1CT.AMU	PVBusLogger
ARMNeoverseN1CT.AMU.mapper	PVBusMapper
ARMNeoverseN1CT.DAP	PVBusLogger
ARMNeoverseN1CT.DAP.mapper	PVBusMapper
ARMNeoverseN1CT.DSU	DSU
ARMNeoverseN1CT.DSU.l3_flusher	AsyncCacheFlushUnit
ARMNeoverseN1CT.DSU.mpam_busslave	PVBusSlave
ARMNeoverseN1CT.DSU.shared_cache	PVCache
ARMNeoverseN1CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseN1CT.DSU.shared_cache.upstream[Y] (where Y = 0-3)	PVBusSlave
ARMNeoverseN1CT.MMAP	PVBusLogger
ARMNeoverseN1CT.MMAP.mapper	PVBusMapper
ARMNeoverseN1CT.RAS	PVBusLogger
ARMNeoverseN1CT.RAS.mapper	PVBusMapper
ARMNeoverseN1CT.cpu0	ARM_Neoverse-N1
ARMNeoverseN1CT.cpu0.UTLB	TLB



Name	Component type
ARMNeoverseN1CT.cpu0.l1dcache	PVCache
ARMNeoverseN1CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseN1CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMNeoverseN1CT.cpu0.l1icache	PVCache
ARMNeoverseN1CT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseN1CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMNeoverseN1CT.cpu0.l2cache	PVCache
ARMNeoverseN1CT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseN1CT.cpu0.l2cache.upstream[U] (where U = 0–1)	PVBusSlave
ARMNeoverseN1CT.ext_bus	PVBusLogger
ARMNeoverseN1CT.ext_bus.mapper	PVBusMapper
ARMNeoverseN1CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

### Ports for ARMNeoverseN1CT

Port	Direction	Protocol	Description
acp_s	slave	PVBus	AXI ACP slave port.
AENDMP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
ASTARTMP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	slave	Signal	CHI defined pins.
broadcastcachemaint	slave	Signal	ACE defined pins.
broadcastouter	slave	Signal	ACE defined pins.
broadcastpersist	slave	Signal	CHI defined pins.
cfgend	slave	Signal	This signal is for EE bit initialisation.
cfgsdisable	slave	Signal	This signal disables write access to some secure Interrupt Controller registers.
cfgte	slave	Signal	This signal provides default exception handling state.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
clusterid	slave	Value	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set ManagerID64 of the core.
clusterpmuirq	master	Signal	DynalIQ pmu irq
CNTHPIRQ	master	Signal	Timer signals to SOC.
CNTHVIRQ	master	Signal	Timer signals to SOC.

Port	Direction	Protocol	Description
CNTPNSIRQ	master	Signal	Timer signals to SOC.
CNTPSIRQ	master	Signal	Timer signals to SOC.
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.
CNTVIRQ	master	Signal	Timer signals to SOC.
commirq	master	Signal	Interrupt signal from debug communications channel.
core_clk_in	slave	ClockSignal	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
cpuporeset	slave	Signal	CPU power on reset. Initializes all the processor logic, including debug logic.
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
ctidbgirq	master	Signal	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	slave	Signal	External debug interface.
dbgnopwrdown	master	Signal	No power-down request.
dbgpwrupreq	master	Signal	Debug power up request.
dev_debug_s	slave	PVBus	External debug interface.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	slave	PVBus	External debug interface.
niden	slave	Signal	External debug interface.
pchannel_cluster	slave	PChannel	PChannel for cluster.
pchannel_core	slave	PChannel	PChannels for cores
pmbirq	master	Signal	Interrupt signal from the statistical profiling unit.
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	master	PVBus	The core will generate bus requests on this port.
pvbus_periph_m	master	PVBus	The core can generate peripheral bus request on this port.
rei	slave	Signal	Per core RAM Error Interrupt.
reset	slave	Signal	Raising this signal will put the core into reset mode.
rvbaraddr	slave	Value_64	Reset vector base address.
sci_m	master	SystemCoherencyInterface	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei	slave	Signal	Per core System Error physical pins.
spiden	slave	Signal	External debug interface.
spniden	slave	Signal	External debug interface.

Port	Direction	Protocol	Description
sporeset	slave	Signal	A single cluster-wide power on reset signal for all resettable registers in DynamIQ.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfirq	slave	Signal	Virtualised FIQ.
vinithi	slave	Signal	This signal controls of the location of the exception vectors at reset.
virq	slave	Signal	Virtualised IRQ.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Per core virtual System Error physical pins.

## Parameters for ARMNeoverseN1CT

### **cpuX.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **cpuX.CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **cpuX.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **cpuX.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**cpuX.VINITHI**

Reset value of SCTLR.V.

Type: `bool`

Default value: `false`

**cpuX.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**cpuX.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**cpuX.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**cpuX.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**cpuX.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**cpuX.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**cpuX.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**cpuX.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**cpuX.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**cpuX.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**cpuX.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**cpuX.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**cpuX.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

### **`cpuX.semihosting-cmd_line`**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

### **`cpuX.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **`cpuX.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`cpuX.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`cpuX.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`cpuX.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

### **`cpuX.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`cpuX.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **BROADCASTATOMIC**

Enable broadcasting of atomic operation. The `broadcastatomic` signal will override this value if used.

Type: `bool`

Default value: `true`

### **BROADCASTCACHEMAINT**

Enable broadcasting of cache maintenance operations to downstream caches. The `broadcastcachemaint` signal will override this value if used.

Type: `bool`

Default value: `true`

### **BROADCASTOUTER**

Enable broadcasting of Outer Shareable transactions. The `broadcastouter` signal will override this value if used.

Type: `bool`

Default value: `true`

### **BROADCASTPERSIST**

Enable broadcasting of cache clean to the point of persistence operations. The `broadcastpersist` signal will override this value if used.



Type: `bool`

Default value: `true`

### **CLUSTER\_ID**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the ManagerID64 of the core.

Type: `uint32_t`

Default value: `0x0`

### **CMO\_broadcast\_when\_cache\_state\_modelling\_disabled**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. This value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required.

Type: `uint8_t`

Default value: `1`

### **GICDISABLE**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`

Default value: `true`

### **NUM\_CORES**

Number of cores per cluster.

Type: `uint8_t`

Default value: `1`

### **cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: 0x1

**cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: 0x1

**dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**dcache-maintenance\_latency**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**dcache-miss\_latency**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**dcache-prefetch\_enabled**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

Type: `bool`

Default value: `false`

**dcache-read\_access\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **dcache-state\_modelled**

Set whether D-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **dcache-write\_access\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**default\_opmode**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

Type: `uint8_t`

Default value: 4

**diagnostics**

Enable DynamIQ diagnostic messages.

Type: `bool`

Default value: `false`

**enable\_simulation\_performance\_optimizations**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of `stage12_tlb_size` parameter to 1024).

Type: `bool`

Default value: `true`

**ext\_abort\_device\_read\_is\_sync**

Synchronous reporting of device-nGnRE read external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_device\_write\_is\_sync**

Synchronous reporting of device-nGnRE write external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_so\_read\_is\_sync**

Synchronous reporting of device-nGnRnE read external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_so\_write\_is\_sync**

Synchronous reporting of device-nGnRnE write external aborts.

Type: `bool`

Default value: `false`

**force\_zero\_PSTATE\_PAN**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

**has\_acp**

If true, Accelerator Coherency Port is configured.

Type: `bool`

Default value: `false`

**has\_coherent\_icache**

Whether icache invalidation to the point of unification is required for instruction to data coherence.  
`true` - Invalidate operations not required.

Type: `bool`

Default value: `false`

**has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**has\_peripheral\_port**

If true, additional AXI peripheral port is configured.

Type: `bool`

Default value: `false`

### **has\_statistical\_profiling**

Whether Statistical Based Profiling is implemented (FEAT\_SPE).

Type: `bool`

Default value: `true`

### **icache-hit\_latency**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

### **icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and

intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`icache-read_latency`**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`icache-state_modelled`**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **`invalidate_code_cache_on_icache_cmo`**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: `uint8_t`

Default value: `0`

### **`l3cache-hit_latency`**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-maintenance_latency`**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-miss\_latency**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-read\_access\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size `l3cache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-read\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l3cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-size**

L3 Cache size in bytes.

Type: `uint32_t`

Default value: `0x100000`

**l3cache-snoop\_data\_transfer\_latency**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-snoop\_issue\_latency**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.



Type: `uint64_t`

Default value: `0x0`

### **`l3cache-write_access_latency`**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size `l3cache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-write_latency`**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l3cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`pchannel_treat_simreset_as_poreset`**

Register core as ON state to cluster with simulation reset.

Type: `bool`

Default value: `false`

### **`periph_address_end`**

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

Type: `uint64_t`

Default value: `0x0`

### **`periph_address_start`**

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

Type: `uint64_t`

Default value: `0x0`

### **`ptw_latency`**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

### **tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

### **tlbi\_stall\_enabled**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: bool

Default value: false

### **treat-dcache-cmos-to-pou-as-nop**

Whether dcache invalidation to the point of unification is required for instruction to data coherence. 0 - Invalidate ops required, 1 - Invalidate ops not required and cannot generate faults, 2 - Invalidate ops not required but can generate faults.

Type: uint8\_t

Default value: 0

### **walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

## 3.106 ARMNeoverseN2CT

Defined in LISA/ARMNeoverseN2CT.lisa.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following parameters were added:

- `has_delayed_dbgreg`
- `has_delayed_sysreg`

## About ARMNeoverseN2CT

The model supports the following features:

- DynamIQ (DSU) system registers.
- Per-core L2 cache.
- A P-Channel for the cluster and for each core.
- Optional peripheral port.
- L3 cache partition.
- Per-core clock.
- `BROADCASTPERSIST` pin.

Support for the following features is planned for a future release:

- DynamIQ Shared Unit-110 (DSU-110) system registers.
- `BROADCASTCACHEMAINTPOU` pin
- `COREINSTRRET`, `COREINSTRRUN`, and `nPMBIRQ` signals
- DSU-110 cluster. The implementation relies on DynamIQ only.
- TRBE.

The following features will not be implemented:

- 256-bit wide output transactions.
- Error correction/detection features.
- Self-test features (MBIST).
- Snoop filtering.
- Armv9 trace extensions.

This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (`crypto.dll` or `crypto.so`) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

## Iris and MTI instances for ARMNeoverseN2CT

This model has the following Iris instances:

Name	Instance type
ARMNeoverseN2CT	Cluster_ARM_Neoverse-N2
ARMNeoverseN2CT.AMU	PVBusLogger
ARMNeoverseN2CT.AMU.mapper	PVBusMapper
ARMNeoverseN2CT.DAP	PVBusLogger

Name	Instance type
ARMNeoverseN2CT.DAP.mapper	PVBusMapper
ARMNeoverseN2CT.DSU	DSU-110
ARMNeoverseN2CT.DSU.PPU_cluster	PPUv1
ARMNeoverseN2CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMNeoverseN2CT.DSU.PPU_core0	PPUv1
ARMNeoverseN2CT.DSU.PPU_core0.busslave	PVBusSlave
ARMNeoverseN2CT.DSU.mpam_busslave	PVBusSlave
ARMNeoverseN2CT.DSU.shared_cache	PVCache
ARMNeoverseN2CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseN2CT.DSU.shared_cache.upstream[Y] (where Y = 0-3)	PVBusSlave
ARMNeoverseN2CT.DSU.utility_slave[0]	PVBusSlave
ARMNeoverseN2CT.MMAP	PVBusLogger
ARMNeoverseN2CT.MMAP.mapper	PVBusMapper
ARMNeoverseN2CT.RAS	PVBusLogger
ARMNeoverseN2CT.RAS.mapper	PVBusMapper
ARMNeoverseN2CT.cpu0	ARM_Neoverse-N2
ARMNeoverseN2CT.cpu0.UTLB	TLB
ARMNeoverseN2CT.cpu0.debug_rom	debug_rom
ARMNeoverseN2CT.cpu0.dtlb	TLB
ARMNeoverseN2CT.cpu0.l1dcache	PVCache
ARMNeoverseN2CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseN2CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMNeoverseN2CT.cpu0.l1icache	PVCache
ARMNeoverseN2CT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseN2CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMNeoverseN2CT.cpu0.l2cache	PVCache
ARMNeoverseN2CT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseN2CT.cpu0.l2cache.upstream[U] (where U = 0-1)	PVBusSlave
ARMNeoverseN2CT.default_CTM	EmbeddedCT
ARMNeoverseN2CT.ext_bus	PVBusLogger
ARMNeoverseN2CT.ext_bus.mapper	PVBusMapper
ARMNeoverseN2CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMNeoverseN2CT.global_debug_rom	debug_rom
ARMNeoverseN2CT.secondary_debug_rom	debug_rom
ARMNeoverseN2CT.sve	ScalableVectorExtension

This model has the following MTI trace components:

Name	Component type
ARMNeoverseN2CT.AMU	PVBusLogger
ARMNeoverseN2CT.AMU.mapper	PVBusMapper

Name	Component type
ARMNeoverseN2CT.DAP	PVBusLogger
ARMNeoverseN2CT.DAP.mapper	PVBusMapper
ARMNeoverseN2CT.DSU	DSU-110
ARMNeoverseN2CT.DSU.PPU_cluster	PPUv1
ARMNeoverseN2CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMNeoverseN2CT.DSU.PPU_core0	PPUv1
ARMNeoverseN2CT.DSU.PPU_core0.busslave	PVBusSlave
ARMNeoverseN2CT.DSU.mpam_busslave	PVBusSlave
ARMNeoverseN2CT.DSU.shared_cache	PVCache
ARMNeoverseN2CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseN2CT.DSU.shared_cache.upstream[Y] (where Y = 0-3)	PVBusSlave
ARMNeoverseN2CT.DSU.utility_slave[0]	PVBusSlave
ARMNeoverseN2CT.MMAP	PVBusLogger
ARMNeoverseN2CT.MMAP.mapper	PVBusMapper
ARMNeoverseN2CT.RAS	PVBusLogger
ARMNeoverseN2CT.RAS.mapper	PVBusMapper
ARMNeoverseN2CT.cpu0	ARM_Neoverse-N2
ARMNeoverseN2CT.cpu0.UTLB	TLB
ARMNeoverseN2CT.cpu0.l1dcache	PVCache
ARMNeoverseN2CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseN2CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMNeoverseN2CT.cpu0.l1icache	PVCache
ARMNeoverseN2CT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseN2CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMNeoverseN2CT.cpu0.l2cache	PVCache
ARMNeoverseN2CT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseN2CT.cpu0.l2cache.upstream[U] (where U = 0-1)	PVBusSlave
ARMNeoverseN2CT.ext_bus	PVBusLogger
ARMNeoverseN2CT.ext_bus.mapper	PVBusMapper
ARMNeoverseN2CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

## Ports for ARMNeoverseN2CT

Port	Direction	Protocol	Description
acp_s	slave	PVBus	AXI ACP slave port.
AEND0MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND2MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).

Port	Direction	Protocol	Description
AEND3MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART2MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	slave	Signal	CHI defined pins.
broadcastcachemaint	slave	Signal	ACE defined pins.
broadcastouter	slave	Signal	ACE defined pins.
broadcastpersist	slave	Signal	CHI defined pins.
cfgend	slave	Signal	This signal is for EE bit initialisation.
cfgsdisable	slave	Signal	This signal disables write access to some secure Interrupt Controller registers.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcsn_pchannel	master	PChannel	Cluster PCSM signal
cluster_powerdown_out	master	Signal	DEPRECATED - An output from PPU that informs thermal controller of the cluster power info
cluster_ppu_hw_stat	master	Value	Cluster PPU power state
clusterid	slave	Value	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set ManagerID64 of the core.
clusterpmuirq	master	Signal	DynamiQ pmu irq
CNTHPIRQ	master	Signal	Timer signals to SOC.
CNTHPSIRQ	master	Signal	Timer signals to SOC
CNTHVIRQ	master	Signal	Timer signals to SOC.
CNTHVSIQ	master	Signal	Timer signals to SOC
CNTPNSIRQ	master	Signal	Timer signals to SOC.
CNTPSIQ	master	Signal	Timer signals to SOC.
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.
CNTVIRQ	master	Signal	Timer signals to SOC.
commirq	master	Signal	Interrupt signal from debug communications channel.

Port	Direction	Protocol	Description
core_clk_in	slave	ClockSignal	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
core_pcsn_pchannel	master	PChannel	Core PCSM signals
core_powerdown_out	master	Signal	DEPRECATED - An output from PPU that informs thermal controller of the core power info
core_ppu_hw_stat	master	Value	Notify the power state change inside the PPU.
coreerrirq	master	Signal	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq	master	Signal	Fault indicator for a detected 1 or 2 bit ECC error
coreinstrrun	master	Signal	Core Running State
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
ctidbgirq	master	Signal	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	slave	Signal	External debug interface.
dbgnopwrdown	master	Signal	No power-down request.
dbgpwrupreq	master	Signal	Debug power up request.
defaultMP	slave	Signal	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main master interface ports.
dev_debug_s	slave	PVBus	External debug interface.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.
gicreset	master	Signal	An output from PPU that can be used to reset GICCLK domain components
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	slave	PVBus	External debug interface.
pmbirq	master	Signal	Interrupt signal from the statistical profiling unit.
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	master	Signal	PPU cluster interrupt.
ppu_cluster_isolate	master	Signal	PPU Cluster Domain isolation control.
ppu_cluster_wakerequest	slave	Signal	PPU cluster wake request signal.
ppu_core_irq	master	Signal	PPU core interrupt.
ppu_core_isolate	master	Signal	PPU Core Domain isolation control.
ppu_core_wakerequest	slave	Signal	PPU Core wake request signals.
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	master	PVBus	The core will generate bus requests on this port.

Port	Direction	Protocol	Description
pvbus_periph_m	master	PVBus	The core can generate peripheral bus request on this port.
rei	slave	Signal	Per core RAM Error Interrupt.
reset	slave	Signal	Reset signal to cluster.
rvbaraddr	slave	Value_64	Reset vector base address.
sci_m	master	SystemCoherencyInterface	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei	slave	Signal	Per core System Error physical pins.
spiden	slave	Signal	External debug interface.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq	master	Signal	Interrupt signal from the trace buffer unit.
utility_bus_s	slave	PVBus	Utility bus slave
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfiq	slave	Signal	Virtualised FIQ.
virq	slave	Signal	Virtualised IRQ.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Per core virtual System Error physical pins.

## Parameters for ARMNeoverseN2CT

### cpu0 . CFGEND

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### cpu0 . CRYPTODISABLE

Disable cryptographic features.

Type: `bool`

Default value: `false`

### cpu0 . RVBARADDR

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`



**cpu0.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter trace\_special\_hlt\_imm16.

Type: bool

Default value: false

**cpu0.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpu0.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpu0.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpu0.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpu0.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **cpu0.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x100000

### **cpu0.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **cpu0.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **cpu0.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **cpu0.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **cpu0.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

**cpu0.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

**cpu0.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**cpu0.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**cpu0.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**cpu0.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**cpu0.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**cpu0.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**cpu0.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**cpu0.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**cpu0.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**cpu0.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**cpu0.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**cpu0.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**cpu0.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

### **`cpu0.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

### **BROADCASTATOMIC**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`

Default value: `true`

### **BROADCASTCACHEMAINT**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

Type: `bool`

Default value: `true`

### **BROADCASTOUTER**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

Type: `bool`

Default value: `true`

### **BROADCASTPERSIST**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: `bool`

Default value: `true`

### **CLUSTER\_ID**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels,

the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the ManagerID64 of the core.

Type: `uint32_t`

Default value: `0x0`

### **CMO\_broadcast\_when\_cache\_state\_modelling\_disabled**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. This value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required.

Type: `uint8_t`

Default value: `1`

### **GICDISABLE**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`

Default value: `true`

### **NUM\_CORES**

Number of cores per cluster.

Type: `uint8_t`

Default value: `1`

### **core\_power\_on\_by\_default**

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPUs which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

Type: `bool`

Default value: `false`

### **cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: 0x1

### **cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: 0x1

### **dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **dcache-maintenance\_latency**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **dcache-miss\_latency**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **dcache-prefetch\_enabled**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

Type: `bool`

Default value: `false`

### **dcache-read\_access\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **dcache-state\_modelled**

Set whether D-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **dcache-write\_access\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0



**default\_opmode**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

Type: `uint8_t`

Default value: 4

**diagnostics**

Enable DynamIQ diagnostic messages.

Type: `bool`

Default value: `false`

**enable\_simulation\_performance\_optimizations**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of `stage12_tlb_size` parameter to 1024).

Type: `bool`

Default value: `true`

**ete.CLAIMTAGS**

Number of claim tags.

Type: `uint8_t`

Default value: 32

**ete.MAX\_INST\_PER\_Q**

Maximum limit for the number of instructions implied by a Q element.

Type: `uint16_t`

Default value: 0x1

**ete.NumberOfRSPairs**

Number of resource selector pairs.

Type: `uint8_t`

Default value: 8

**ete.PIDR\_CM0D**

TRCPIDR CM0D value.

Type: `uint8_t`

Default value: 0

**ete.PIDR\_REVAND**

TRCPIDR REVAND value.

Type: `uint8_t`

Default value: 0

**ete.PIDR\_REVISION**

TRCPIDR REVISION value.

Type: `uint8_t`

Default value: 0

**ete.Q\_CADENCE**

Number of instruction blocks traced between two Q elements.

Type: `uint16_t`

Default value: 0x1

**ete.RES0\_STATEFUL**

Whether **RES0** bits are stateful or **RAZ/WI**.

Type: `bool`

Default value: `false`

**ete.RETSTACK**

Return stack depth.

Type: `uint8_t`

Default value: 3

**ete.REVISION**

TRCIDR1 revision value.

Type: `uint8_t`

Default value: 0

**ete.SIM\_OVERFLOW\_GRANULARITY**

Number of instruction blocks in each granule, for simulated overflow.

Type: `uint32_t`

Default value: `0x64`

**ete.SIM\_OVERFLOW\_PERCENTAGE**

Percentage of instruction blocks lost in each granule, for simulated overflow.

Type: `uint8_t`

Default value: `0`

**ete.SOURCE\_ADDRESS**

Allow generation of source address elements.

Type: `bool`

Default value: `false`

**ete.TRACE\_OUTPUT**

File to which to write trace byte stream.

Type: `string`

Default value: `N/A`

**ete.TRCSRTA\_FORCED\_EXCEP**

TRCSRTA value for a forcibly traced exception.

Type: `bool`

Default value: `false`

**ext\_abort\_device\_read\_is\_sync**

Synchronous reporting of device-nGnRE read external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_device\_write\_is\_sync**

Synchronous reporting of device-nGnRE write external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_so\_read\_is\_sync**

Synchronous reporting of device-nGnRnE read external aborts.

Type: `bool`

Default value: `false`

**ext\_abort\_so\_read\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `uint8_t`

Default value: 0

**ext\_abort\_so\_write\_is\_sync**

Synchronous reporting of device-nGnRnE write external aborts.

Type: `bool`

Default value: `false`

**force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks**

Force MTE tag accesses to **RAZ/WI** and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

Type: `bool`

Default value: `false`

**force\_zero\_PSTATE\_PAN**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

**force\_zero\_mpam\_partid\_and\_pmg**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

Type: `uint8_t`

Default value: 0

**has\_acp**

If true, Accelerator Coherency Port is configured.

Type: `bool`

Default value: `false`

**has\_coherent\_icache**

Whether icache invalidation to the point of unification is required for instruction to data coherence.  
`true` - Invalidate operations not required.

Type: `bool`

Default value: `false`

**has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

**has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**has\_ete**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (`-plugin` or `-P`).

Type: `bool`

Default value: `false`

**has\_external\_rndr**

Implement external random number generator module. When enabling this with `has_rndr` enabled, the external random number generator will be used instead of internal random number generator.

Type: `uint8_t`

Default value: `1`

**has\_peripheral\_port**

If true, an additional AXI peripheral port is configured.

Type: `bool`

Default value: `false`

### **has\_rndr**

Implement random number instructions to read from RNDR and RNDRSS random number registers from ARMv8.5 (FEAT\_RNG).

Type: `uint8_t`

Default value: `1`

### **icache-hit\_latency**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

### **icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and

intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`icache-read_latency`**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`icache-state_modelled`**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **`invalidate_code_cache_on_icache_cmo`**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: `uint8_t`

Default value: `0`

### **`l3cache-hit_latency`**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`l3cache-maintenance_latency`**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-miss\_latency**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-read\_access\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size `l3cache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-read\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l3cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-snoop\_data\_transfer\_latency**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**l3cache-snoop\_issue\_latency**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`



**l3cache-write\_access\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size l3cache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**l3cache-write\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. l3cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**memory\_tagging\_support\_level**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).

Type: uint8\_t

Default value: 2

**mpam\_max\_partid**

MPAM Maximum PARTID Supported.

Type: uint16\_t

Default value: 511

**mpmm\_accumulator\_multiplier**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator.value of n means the accumulator will use (n \* accumulator value) to calculate the mpmm threshold (MPMM).is provided as a fast model workaround to handle cases where execution of quantum in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

Type: uint8\_t

Default value: 1

**periph\_address\_end**

End address for peripheral port address range exclusive(corresponds to AENDMP input signal).

Type: `uint64_t`

Default value: `0x0`

### **`periph_address_start`**

Start address for peripheral port address range inclusive (corresponds to ASTARTMP input signal).

Type: `uint64_t`

Default value: `0x0`

### **`ptw_latency`**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

### **`rndr_rndrrs_seed`**

Initial seed for random engine used in RNDR register.

Type: `uint64_t`

Default value: `0x0`

### **`tlb_latency`**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

### **`tlbi_stall_enabled`**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: `bool`

Default value: `false`

### **`treat-dcache-cmos-to-pou-as-nop`**

Whether dcache invalidation to the point of unification is required for instruction to data coherence. 0 - Invalidate ops required, 1 - Invalidate ops not required and cannot generate faults, 2 - Invalidate ops not required but can generate faults.

Type: `uint8_t`

Default value: `0`

**walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

## 3.107 ARMNeoverseN3CT

Defined in `LISA/ARMNeoverseN3CT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following parameters were added:

- `has_delayed_dbgreg`
- `has_delayed_sysreg`

### About ARMNeoverseN3CT

A DSU-120 DynamIQ cluster containing a single Neoverse-N3 core configured for Direct connect.

The following DSU/CPU features will not be implemented:

- DynamIQ features that are irrelevant to the programmers' view simulation.
- 256-bit wide output transactions.
- ECC and parity schemes are hardware-specific so are not supported.
- Self-test features (MBIST).
- Snoop filtering.
- Cache stashing capability.
- Embedded Logic Analyzer (ELA).

This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (`crypto.dll` or `crypto.so`) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

### Iris and MTI instances for ARMNeoverseN3CT

This model has the following Iris instances:

Name	Instance type
ARMNeoverseN3CT	Cluster_ARM_Neoverse-N3
ARMNeoverseN3CT.AMU	PVBusLogger
ARMNeoverseN3CT.AMU.mapper	PVBusMapper
ARMNeoverseN3CT.DAP	PVBusLogger
ARMNeoverseN3CT.DAP.mapper	PVBusMapper
ARMNeoverseN3CT.DSU	DSU-120
ARMNeoverseN3CT.DSU.PPU_cluster	PPUv1
ARMNeoverseN3CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMNeoverseN3CT.DSU.PPU_core0	PPUv1
ARMNeoverseN3CT.DSU.PPU_core0.busslave	PVBusSlave
ARMNeoverseN3CT.DSU.mpam_busslave	PVBusSlave
ARMNeoverseN3CT.DSU.shared_cache	PVCache
ARMNeoverseN3CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseN3CT.DSU.shared_cache.upstream[Y] (where Y = 0-4)	PVBusSlave
ARMNeoverseN3CT.DSU.utility_slave[0]	PVBusSlave
ARMNeoverseN3CT.MMAP	PVBusLogger
ARMNeoverseN3CT.MMAP.mapper	PVBusMapper
ARMNeoverseN3CT.RAS	PVBusLogger
ARMNeoverseN3CT.RAS.mapper	PVBusMapper
ARMNeoverseN3CT.cpu0	ARM_Neoverse-N3
ARMNeoverseN3CT.cpu0.UTLB	TLB
ARMNeoverseN3CT.cpu0.debug_rom	debug_rom
ARMNeoverseN3CT.cpu0.dtlb	TLB
ARMNeoverseN3CT.cpu0.l1dcache	PVCache
ARMNeoverseN3CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseN3CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMNeoverseN3CT.cpu0.l1icache	PVCache
ARMNeoverseN3CT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseN3CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMNeoverseN3CT.cpu0.l2cache	PVCache
ARMNeoverseN3CT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseN3CT.cpu0.l2cache.upstream[U] (where U = 0-1)	PVBusSlave
ARMNeoverseN3CT.default_CTM	EmbeddedCT
ARMNeoverseN3CT.ext_bus	PVBusLogger
ARMNeoverseN3CT.ext_bus.mapper	PVBusMapper
ARMNeoverseN3CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMNeoverseN3CT.global_debug_rom	debug_rom
ARMNeoverseN3CT.secondary_debug_rom	debug_rom
ARMNeoverseN3CT.sve	ScalableVectorExtension

This model has the following MTI trace components:

Name	Component type
ARMNeoverseN3CT.AMU	PVBusLogger
ARMNeoverseN3CT.AMU.mapper	PVBusMapper
ARMNeoverseN3CT.DAP	PVBusLogger
ARMNeoverseN3CT.DAP.mapper	PVBusMapper
ARMNeoverseN3CT.DSU	DSU-120
ARMNeoverseN3CT.DSU.PPU_cluster	PPUv1
ARMNeoverseN3CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMNeoverseN3CT.DSU.PPU_core0	PPUv1
ARMNeoverseN3CT.DSU.PPU_core0.busslave	PVBusSlave
ARMNeoverseN3CT.DSU.mpam_busslave	PVBusSlave
ARMNeoverseN3CT.DSU.shared_cache	PVCache
ARMNeoverseN3CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseN3CT.DSU.shared_cache.upstream[Y] (where Y = 0-4)	PVBusSlave
ARMNeoverseN3CT.DSU.utility_slave[0]	PVBusSlave
ARMNeoverseN3CT.MMAP	PVBusLogger
ARMNeoverseN3CT.MMAP.mapper	PVBusMapper
ARMNeoverseN3CT.RAS	PVBusLogger
ARMNeoverseN3CT.RAS.mapper	PVBusMapper
ARMNeoverseN3CT.cpu0	ARM_Neoverse-N3
ARMNeoverseN3CT.cpu0.UTLB	TLB
ARMNeoverseN3CT.cpu0.l1dcache	PVCache
ARMNeoverseN3CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseN3CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMNeoverseN3CT.cpu0.l1icache	PVCache
ARMNeoverseN3CT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseN3CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMNeoverseN3CT.cpu0.l2cache	PVCache
ARMNeoverseN3CT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseN3CT.cpu0.l2cache.upstream[U] (where U = 0-1)	PVBusSlave
ARMNeoverseN3CT.ext_bus	PVBusLogger
ARMNeoverseN3CT.ext_bus.mapper	PVBusMapper
ARMNeoverseN3CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

### Ports for ARMNeoverseN3CT

Port	Direction	Protocol	Description
acp_s	slave	PVBus	AXI ACP slave port.
AEND0MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).

Port	Direction	Protocol	Description
AEND2MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND3MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART2MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	slave	Signal	CHI defined pins.
broadcastcachemaint	slave	Signal	ACE defined pins.
broadcastouter	slave	Signal	ACE defined pins.
broadcastpersist	slave	Signal	CHI defined pins.
cfgend	slave	Signal	This signal is for EE bit initialisation.
cfgsdisable	slave	Signal	This signal disables write access to some secure Interrupt Controller registers.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcsmpchannel	master	PChannel	Cluster PCSM signal
cluster_powerdown_out	master	Signal	DEPRECATED - An output from PPU that informs thermal controller of the cluster power info
cluster_ppu_hw_stat	master	Value	Cluster PPU power state
clusterid	slave	Value	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set ManagerID64 of the core.
clusterpmuirq	master	Signal	DynamiQ pmu irq
CNTHPIRQ	master	Signal	Timer signals to SOC.
CNTHPSIRQ	master	Signal	Timer signals to SOC
CNTHVIRQ	master	Signal	Timer signals to SOC.
CNTHVSIRQ	master	Signal	Timer signals to SOC
CNTPNSIRQ	master	Signal	Timer signals to SOC.
CNTPSIRQ	master	Signal	Timer signals to SOC.
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.

Port	Direction	Protocol	Description
CNTVIRQ	master	Signal	Timer signals to SOC.
commirq	master	Signal	Interrupt signal from debug communications channel.
core_clk_in	slave	ClockSignal	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
core_pcsn_pchannel	master	PChannel	Core PCSM signals
core_powerdown_out	master	Signal	DEPRECATED - An output from PPUs that informs thermal controller of the core power info
core_ppu_hw_stat	master	Value	Notify the power state change inside the PPU.
coreerrirq	master	Signal	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq	master	Signal	Fault indicator for a detected 1 or 2 bit ECC error
coreinstrrun	master	Signal	Core Running State
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
ctidbgirq	master	Signal	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	slave	Signal	External debug interface.
dbgnopwrdown	master	Signal	No power-down request.
dbgpwrupreq	master	Signal	Debug power up request.
defaultMP	slave	Signal	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main master interface ports.
dev_debug_s	slave	PVBus	External debug interface.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.
gicreset	master	Signal	An output from PPUs that can be used to reset GICCLK domain components
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	slave	PVBus	External debug interface.
pmbirq	master	Signal	Interrupt signal from the statistical profiling unit.
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	master	Signal	PPU cluster interrupt.
ppu_cluster_isolate	master	Signal	PPU Cluster Domain isolation control.
ppu_cluster_wakerequest	slave	Signal	PPU cluster wake request signal.
ppu_core_irq	master	Signal	PPU core interrupt.
ppu_core_isolate	master	Signal	PPU Core Domain isolation control.
ppu_core_wakerequest	slave	Signal	PPU core wake request signal.
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.

Port	Direction	Protocol	Description
pvbus_m0	master	PVBus	The core will generate bus requests on this port.
pvbus_periph_m	master	PVBus	The core can generate peripheral bus request on this port.
rei	slave	Signal	Per core RAM Error Interrupt.
reset	slave	Signal	Reset signal to cluster.
rvbaraddr	slave	Value_64	Reset vector base address.
sci_m	master	SystemCoherencyInterface	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei	slave	Signal	Per core System Error physical pins.
spiden	slave	Signal	External debug interface.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq	master	Signal	Interrupt signal from the trace buffer unit.
utility_bus_s	slave	PVBus	Utility bus slave
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfiq	slave	Signal	Virtualised FIQ.
virq	slave	Signal	Virtualised IRQ.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Per core virtual System Error physical pins.

## Parameters for ARMNeoverseN3CT

### cpu0 . CFGEND

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### cpu0 . CFGTE

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### cpu0 . CRYPTODISABLE

Disable cryptographic features.

Type: `bool`



Default value: `false`

**cpu0.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**cpu0.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**cpu0.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

**cpu0.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**cpu0.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**cpu0.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**cpu0.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpu0.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpu0.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x100000

**cpu0.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpu0.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpu0.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

### **`cpu0.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`cpu0.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

### **`cpu0.min_sync_level`**

Force minimum `syncLevel` (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

### **`cpu0.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`cpu0.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`cpu0.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**cpu0.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**cpu0.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**cpu0.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**cpu0.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**cpu0.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**cpu0.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: 0x0

**cpu0.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: 0xf000000

**cpu0.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**cpu0.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**cpu0.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**cpu0.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**cpu0.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**AEND0\_DEFAULT**

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMPO input signal).

Type: `uint64_t`

Default value: `0x0`

**AEND1\_DEFAULT**

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

Type: `uint64_t`

Default value: `0x0`

**AEND2\_DEFAULT**

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

Type: `uint64_t`

Default value: `0x0`

**AEND3\_DEFAULT**

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART0\_DEFAULT**

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART1\_DEFAULT**

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART2\_DEFAULT**

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART3\_DEFAULT**

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

Type: `uint64_t`

Default value: `0x0`

**BROADCASTATOMIC**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`

Default value: `true`

**BROADCASTCACHEMAINT**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

Type: `bool`

Default value: `false`

**BROADCASTOUTER**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

Type: `bool`

Default value: `false`

**BROADCASTPERSIST**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: `bool`

Default value: `true`

**CLUSTER\_ID**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the ManagerID64 of the core.

Type: `uint32_t`

Default value: `0x0`

### **CMO\_broadcast\_when\_cache\_state\_modelling\_disabled**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. This value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required.

Type: `uint8_t`

Default value: `1`

### **CPUCFR**

Value of CPU Configuration Register.

Type: `uint64_t`

Default value: `0x4`

### **GICDISABLE**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`

Default value: `true`

### **NUM\_CORES**

Number of cores per cluster.

Type: `uint8_t`

Default value: `1`

### **bus\_type**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

Type: `uint8_t`

Default value: `0`



**core\_power\_on\_by\_default**

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPUs which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

Type: `bool`

Default value: `false`

**cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-maintenance\_latency**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-miss\_latency**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-prefetch\_enabled**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

Type: `bool`

Default value: `false`

**dcache-read\_access\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**dcache-size**

L1 D-Cache size in bytes.

Type: `uint32_t`

Default value: `0x10000`

**dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**dcache-state\_modelled**

Set whether D-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **`dcache-write_access_latency`**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`dcache-write_latency`**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`default_opmode`**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

Type: `uint8_t`

Default value: 4

### **`diagnostics`**

Enable DynamIQ diagnostic messages.

Type: `bool`

Default value: `false`

### **`enable_simulation_performance_optimizations`**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of `stage12_tlb_size` parameter to 1024).

Type: `bool`

Default value: `true`

### **`ete.CLAIMTAGS`**

Number of claim tags.

Type: `uint8_t`

Default value: 4

### **ete.ETE\_REVISION**

ETE revision: 0=ETEv1.0, 1=ETEv1.1, 2=ETEv1.2, 3=ETEv1.3.

Type: `uint8_t`

Default value: 1

### **ete.MAX\_INST\_PER\_Q**

Maximum limit for the number of instructions implied by a Q element.

Type: `uint16_t`

Default value: 0x1

### **ete.NumberOfRSPairs**

Number of resource selector pairs.

Type: `uint8_t`

Default value: 8

### **ete.PIDR\_CM0D**

TRCPIDR CM0D value.

Type: `uint8_t`

Default value: 0

### **ete.PIDR\_REVAND**

TRCPIDR REVAND value.

Type: `uint8_t`

Default value: 0

### **ete.PIDR\_REVISION**

TRCPIDR REVISION value.

Type: `uint8_t`

Default value: 0

### **ete.Q\_CADENCE**

Number of instruction blocks traced between two Q elements.

Type: `uint16_t`

Default value: `0x1`

### **`ete.RES0_STATEFUL`**

Whether **RES0** bits are stateful or **RAZ/WI**.

Type: `bool`

Default value: `false`

### **`ete.RETSTACK`**

Return stack depth.

Type: `uint8_t`

Default value: `1`

### **`ete.REVISION`**

TRCIDR1 revision value.

Type: `uint8_t`

Default value: `0`

### **`ete.SIM_OVERFLOW_GRANULARITY`**

Number of instruction blocks in each granule, for simulated overflow.

Type: `uint32_t`

Default value: `0x64`

### **`ete.SIM_OVERFLOW_PERCENTAGE`**

Percentage of instruction blocks lost in each granule, for simulated overflow.

Type: `uint8_t`

Default value: `0`

### **`ete.SOURCE_ADDRESS`**

Allow generation of source address elements.

Type: `bool`

Default value: `false`

### **`ete.TRACE_OUTPUT`**

File to which to write trace byte stream.

Type: `string`

Default value: `N/A`

### **`ete.TRCRSRTA_FORCED_EXCEP`**

TRCRSR.TA value for a forcibly traced exception.

Type: `bool`

Default value: `false`

### **`ete.TSMARK`**

Whether timestamp markers are supported.

Type: `bool`

Default value: `true`

### **`force_mte_tag_access_razwi_and_ignore_tag_checks`**

Force MTE tag accesses to **RAZ/WI** and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

Type: `bool`

Default value: `false`

### **`force_zero_PSTATE_PAN`**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

### **`force_zero_mpam_partid_and_pmg`**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAMO\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

Type: `uint8_t`

Default value: `0`

### **`has_coherent_icache`**

Whether icache invalidation to the point of unification is required for instruction to data coherence.  
`true` - Invalidate operations not required.

Type: `bool`

Default value: `true`

### **has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

### **has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

### **has\_enhanced\_pan**

Implements Armv8.7 Enhanced PAN feature (FEAT\_PAN3) values of this parameter are:- 1, feature is implemented if ARMv8.7 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 2

### **has\_ete**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (-plugin or -P).

Type: `bool`

Default value: `false`

### **has\_large\_va**

Implement support for the extended 52-bit virtual addresses from ARMv8.2 (FEAT\_LVA). values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.2 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

### **has\_peripheral\_port**

If true, an additional AXI peripheral port is configured.

Type: `bool`

Default value: `false`

### **has\_rndr\_trap**

Implement trapping for RNDR and RNDRSS random number registers from ARMv8.8. (FEAT\_RNG\_TRAP) values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.8 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 2

### **has\_statistical\_profiling**

Whether Statistical Based Profiling is implemented (FEAT\_SPE).

Type: `bool`

Default value: `false`

### **icache-hit\_latency**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`



Default value: `false`

### **`icache-read_access_latency`**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`icache-read_latency`**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`icache-size`**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: `0x10000`

### **`icache-state_modelled`**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **`instruction_tlb_size`**

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

Type: `uint32_t`

Default value: `0x0`

### **`invalidate_code_cache_on_icache_cmo`**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: `uint8_t`

Default value: 0

### **memory\_tagging\_support\_level**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

Type: `uint8_t`

Default value: 3

### **mpmm\_accumulator\_multiplier**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator.value of n means the accumulator will use (n \* accumulator value) to calculate the mpmm threshold (MPMM).is provided as a fast model workaround to handle cases where execution of quantum in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

Type: `uint8_t`

Default value: 1

### **pmu-num\_counters**

Number of PMU counters implemented.

Type: `uint8_t`

Default value: 6

### **ptw\_latency**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

### **stage12\_tlb\_size**

Number of stage1+2 tlb entries. Valid values are 0 or >= 4.

Type: `uint32_t`

Default value: 0x80

### **tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

### **`tlbi_stall_enabled`**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: `bool`

Default value: `false`

### **`treat_PAC_as_NOP`**

Non-architecture parameter. Treat Pointer Authentication as **NOP**. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are **NOP**. Brach, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

Type: `bool`

Default value: `false`

### **`walk_cache_latency`**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

## 3.108 ARMNeoverseV1CT

Defined in `LISA/ARMNeoverseV1CT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following parameters were added:

- `has_delayed_dbgreg`
- `has_delayed_sysreg`

## About ARMNeoverseV1CT

The model supports the following features:

- DynamIQ (DSU) system registers.
- Per-core L2 cache.
- A P-Channel for the cluster and for each core.
- Optional peripheral port.
- L3 cache partition.
- Per-core clock.
- BROADCASTPERSIST pin.

Support for the following features is planned for a future release:

- BROADCASTCACHEMAINTPOU pin.
- COREINSTRRET, COREINSTRRUN, and nPMBIRQ signals.
- A common cache that is shared by all threads of the core. Currently, each thread has its own L1 cache and L2 cache.
- Per-thread parameters, although signals are implemented.

The following features will not be implemented:

- 256-bit wide output transactions.
- ECC and parity schemes are hardware-specific so are not supported.
- Self-test features (MBIST).
- Snoop filtering.
- DynamIQ features that are irrelevant to the programmers' view simulation, for example:
  - Automatic CPU retention mode.
  - Level-3 Cache RAM retention.
- Latency configuration.
- Cache stashing capability.

This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (crypto.dll or crypto.so) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

## Iris and MTI instances for ARMNeoverseV1CT

This model has the following Iris instances:

Name	Instance type
ARMNeoverseV1CT	Cluster_ARM_Neoverse-V1
ARMNeoverseV1CT.AMU	PVBusLogger
ARMNeoverseV1CT.AMU.mapper	PVBusMapper

Name	Instance type
ARMNeoverseV1CT.DAP	PVBusLogger
ARMNeoverseV1CT.DAP.mapper	PVBusMapper
ARMNeoverseV1CT.DSU	DSU
ARMNeoverseV1CT.DSU.mpam_busslave	PVBusSlave
ARMNeoverseV1CT.DSU.shared_cache	PVCache
ARMNeoverseV1CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseV1CT.DSU.shared_cache.upstream[Y] (where Y = 0-3)	PVBusSlave
ARMNeoverseV1CT.MMAP	PVBusLogger
ARMNeoverseV1CT.MMAP.mapper	PVBusMapper
ARMNeoverseV1CT.RAS	PVBusLogger
ARMNeoverseV1CT.RAS.mapper	PVBusMapper
ARMNeoverseV1CT.cpu0	ARM_Neoverse-V1
ARMNeoverseV1CT.cpu0.UTLB	TLB
ARMNeoverseV1CT.cpu0.debug_rom	debug_rom
ARMNeoverseV1CT.cpu0.dtlb	TLB
ARMNeoverseV1CT.cpu0.l1dcache	PVCache
ARMNeoverseV1CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseV1CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMNeoverseV1CT.cpu0.l1icache	PVCache
ARMNeoverseV1CT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseV1CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMNeoverseV1CT.cpu0.l2cache	PVCache
ARMNeoverseV1CT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseV1CT.cpu0.l2cache.upstream[U] (where U = 0-1)	PVBusSlave
ARMNeoverseV1CT.ext_bus	PVBusLogger
ARMNeoverseV1CT.ext_bus.mapper	PVBusMapper
ARMNeoverseV1CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMNeoverseV1CT.global_debug_rom	debug_rom
ARMNeoverseV1CT.secondary_debug_rom	debug_rom
ARMNeoverseV1CT.sve	ScalableVectorExtension

This model has the following MTI trace components:

Name	Component type
ARMNeoverseV1CT.AMU	PVBusLogger
ARMNeoverseV1CT.AMU.mapper	PVBusMapper
ARMNeoverseV1CT.DAP	PVBusLogger
ARMNeoverseV1CT.DAP.mapper	PVBusMapper
ARMNeoverseV1CT.DSU	DSU
ARMNeoverseV1CT.DSU.mpam_busslave	PVBusSlave
ARMNeoverseV1CT.DSU.shared_cache	PVCache

Name	Component type
ARMNeoverseV1CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseV1CT.DSU.shared_cache.upstream[Y] (where Y = 0–3)	PVBusSlave
ARMNeoverseV1CT.MMAP	PVBusLogger
ARMNeoverseV1CT.MMAP.mapper	PVBusMapper
ARMNeoverseV1CT.RAS	PVBusLogger
ARMNeoverseV1CT.RAS.mapper	PVBusMapper
ARMNeoverseV1CT.cpu0	ARM_Neoverse-V1
ARMNeoverseV1CT.cpu0.UTLB	TLB
ARMNeoverseV1CT.cpu0.l1dcache	PVCache
ARMNeoverseV1CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseV1CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMNeoverseV1CT.cpu0.l1icache	PVCache
ARMNeoverseV1CT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseV1CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMNeoverseV1CT.cpu0.l2cache	PVCache
ARMNeoverseV1CT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseV1CT.cpu0.l2cache.upstream[U] (where U = 0–1)	PVBusSlave
ARMNeoverseV1CT.ext_bus	PVBusLogger
ARMNeoverseV1CT.ext_bus.mapper	PVBusMapper
ARMNeoverseV1CT.gic_cpuif_decoder_cluster	GLICv3CPUInterfaceDecoder

### Ports for ARMNeoverseV1CT

Port	Direction	Protocol	Description
acp_s	slave	PVBus	AXI ACP slave port.
AENDMP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
ASTARTMP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	slave	Signal	CHI defined pins.
broadcastcachemaint	slave	Signal	ACE defined pins.
broadcastouter	slave	Signal	ACE defined pins.
broadcastpersist	slave	Signal	CHI defined pins.
cfgend	slave	Signal	This signal if for EE bit initialisation.
cfgsdisable	slave	Signal	This signal disables write access to some secure Interrupt Controller registers.
cfgte	slave	Signal	This signal provides default exception handling state.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.

Port	Direction	Protocol	Description
clusterid	slave	Value	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set ManagerID64 of the core.
clusterpmuirq	master	Signal	DynamiQ pmu irq
CNTHPIRQ	master	Signal	Timer signals to SOC.
CNTHVIRQ	master	Signal	Timer signals to SOC.
CNTPNSIRQ	master	Signal	Timer signals to SOC.
CNTPSIRQ	master	Signal	Timer signals to SOC.
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.
CNTVIRQ	master	Signal	Timer signals to SOC.
commirq	master	Signal	Interrupt signal from debug communications channel.
core_clk_in	slave	ClockSignal	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
cpuporeset	slave	Signal	CPU power on reset. Initializes all the processor logic, including debug logic.
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
ctidbgirq	master	Signal	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	slave	Signal	External debug interface.
dbgnopwrdown	master	Signal	No power-down request.
dbgpwrupreq	master	Signal	Debug power up request.
dev_debug_s	slave	PVBus	External debug interface.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	slave	PVBus	External debug interface.
niden	slave	Signal	External debug interface.
pchannel_cluster	slave	PChannel	PChannel for cluster.
pchannel_core	slave	PChannel	PChannels for cores
pmbirq	master	Signal	Interrupt signal from the statistical profiling unit.
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	master	PVBus	The core will generate bus requests on this port.

Port	Direction	Protocol	Description
pvbus_periph_m	master	PVBus	The core can generate peripheral bus request on this port.
rei	slave	Signal	Per core RAM Error Interrupt.
reset	slave	Signal	Raising this signal will put the core into reset mode.
rvbaraddr	slave	Value_64	Reset vector base address.
sci_m	master	SystemCoherencyInterface	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei	slave	Signal	Per core System Error physical pins.
spiden	slave	Signal	External debug interface.
spniden	slave	Signal	External debug interface.
sporeset	slave	Signal	A single cluster-wide power on reset signal for all resettable registers in DynamIQ.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfiq	slave	Signal	Virtualised FIQ.
vinithi	slave	Signal	This signal controls of the location of the exception vectors at reset.
virq	slave	Signal	Virtualised IRQ.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Per core virtual System Error physical pins.

## Parameters for ARMNeoverseV1CT

### cpuX.CFGEND

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### cpuX.CFGTE

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### cpuX.CRYPTODISABLE

Disable cryptographic features.



Type: `bool`

Default value: `false`

### **`cpuX.RVBARADDR`**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.VINITHI`**

Reset value of SCTL.R.V.

Type: `bool`

Default value: `false`

### **`cpuX.enable_trace_special_hlt_imm16`**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **`cpuX.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**cpuX.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

**cpuX.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**cpuX.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`cpuX.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

### **`cpuX.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

### **`cpuX.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`cpuX.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`cpuX.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

**cpuX.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**cpuX.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**cpuX.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**cpuX.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**cpuX.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**cpuX.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: 0x0

**cpuX.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: 0xf000000

**cpuX.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**cpuX.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**cpuX.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**cpuX.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**BROADCASTATOMIC**

Enable broadcasting of atomic operation. The `broadcastatomic` signal will override this value if used.

Type: `bool`

Default value: `true`

**BROADCASTACHEMAINT**

Enable broadcasting of cache maintenance operations to downstream caches. The `broadcastcachemaint` signal will override this value if used.

Type: `bool`

Default value: `true`

**BROADCASTOUTER**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

Type: `bool`

Default value: `true`

**BROADCASTPERSIST**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: `bool`

Default value: `true`

**CLUSTER\_ID**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the ManagerID64 of the core.

Type: `uint32_t`

Default value: `0x0`

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. This value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required.

Type: `uint8_t`

Default value: `1`

**GICDISABLE**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`

Default value: `true`

**NUM\_CORES**

Number of cores per cluster.

Type: `uint8_t`

Default value: 1

**cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: 0x1

**cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: 0x1

**dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**dcache-maintenance\_latency**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**dcache-miss\_latency**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

**dcache-prefetch\_enabled**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

Type: `bool`

Default value: `false`

### **`dcache-read_access_latency`**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`dcache-read_latency`**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`dcache-size`**

L1 D-Cache size in bytes.

Type: `uint32_t`

Default value: `0x10000`

### **`dcache-snoop_data_transfer_latency`**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`dcache-state_modelled`**

Set whether D-cache has stateful implementation.

Type: `bool`

Default value: `false`



**dcache-write\_access\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**default\_opmode**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

Type: uint8\_t

Default value: 4

**diagnostics**

Enable DynamIQ diagnostic messages.

Type: bool

Default value: false

**enable\_simulation\_performance\_optimizations**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of stage12\_tlb\_size parameter to 1024).

Type: bool

Default value: true

**enhanced\_pac2\_level**

Implements Enhanced PAC2 from ARMv8.6 (FEAT\_PAuth2), and PAC enhancements from ARMv9.5 (FEAT\_PAuth\_LR). options 0-3 of this feature are mandatory for ARMv8.6 but can be

cherry-picked to a ARMv8.3(or greater) implementation. FEAT\_FPACCOMBINE is mandatory in the presence of Future Architecture Technologies (FAT). 0: No EnhancedPAC2, 1: EnhancedPAC2 Only (FEAT\_PAuth2), 2: EnhancedPAC2 with FPAC (FEAT\_FPAC), 3: EnhancedPAC2 with FPACCombined (FEAT\_FPACCOMBINE), 4: EnhancedPAC2 with LR signing (FEAT\_PAuth\_LR).

Type: `uint8_t`

Default value: `1`

#### **`ext_abort_device_read_is_sync`**

Synchronous reporting of device-nGnRE read external aborts.

Type: `bool`

Default value: `false`

#### **`ext_abort_device_write_is_sync`**

Synchronous reporting of device-nGnRE write external aborts.

Type: `bool`

Default value: `false`

#### **`ext_abort_so_read_is_sync`**

Synchronous reporting of device-nGnRnE read external aborts.

Type: `bool`

Default value: `false`

#### **`ext_abort_so_write_is_sync`**

Synchronous reporting of device-nGnRnE write external aborts.

Type: `bool`

Default value: `false`

#### **`force_zero_PSTATE_PAN`**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

**force\_zero\_mpam\_partid\_and\_pmg**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

Type: `uint8_t`

Default value: `0`

**has\_acp**

If true, Accelerator Coherency Port is configured.

Type: `bool`

Default value: `false`

**has\_coherent\_icache**

Whether icache invalidation to the point of unification is required for instruction to data coherence.  
`true` - Invalidate operations not required.

Type: `bool`

Default value: `true`

**has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

**has\_external\_rndr**

Implement external random number generator module. When enabling this with `has_rndr` enabled, the external random number generator will be used instead of internal random number generator.

Type: `uint8_t`

Default value: `1`

**has\_peripheral\_port**

If true, additional AXI peripheral port is configured.

Type: `bool`

Default value: `false`

**has\_rndr**

Implement random number instructions to read from RNDR and RNDRSS random number registers from ARMv8.5 (FEAT\_RNG).

Type: `uint8_t`

Default value: `1`

**icache-hit\_latency**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

**icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-size**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: `0x10000`

**icache-state\_modelled**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

**invalidate\_code\_cache\_on\_icache\_cmo**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: `uint8_t`

Default value: `0`

**l3cache-hit\_latency**

L3 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **l3cache-maintenance\_latency**

L3 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **l3cache-miss\_latency**

L3 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **l3cache-read\_access\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per access (of size `l3cache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l3cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **l3cache-read\_latency**

L3 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l3cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **l3cache-size**

L3 Cache size in bytes.

Type: `uint32_t`

Default value: 0x0

**l3cache-snoop\_data\_transfer\_latency**

L3 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**l3cache-snoop\_issue\_latency**

L3 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**l3cache-write\_access\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per access (of size l3cache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l3cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**l3cache-write\_latency**

L3 Cache timing annotation latency for write accesses given in ticks per byte accessed. l3cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

**pchannel\_treat\_simreset\_as\_poreset**

Register core as ON state to cluster with simulation reset.

Type: bool

Default value: false

**periph\_address\_end**

End address for peripheral port address range exclusive (corresponds to AENDMP input signal).

Type: uint64\_t

Default value: 0x0

**periph\_address\_start**

Start address for peripheral port address range inclusive(corresponds to ASTARTMP input signal).

Type: uint64\_t

Default value: 0x0

**ptw\_latency**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

**tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

**tlbi\_stall\_enabled**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: bool

Default value: false

**treat-dcache-cmos-to-pou-as-nop**

Whether dcache invalidation to the point of unification is required for instruction to data coherence. 0 - Invalidate ops required, 1 - Invalidate ops not required and cannot generate faults, 2 - Invalidate ops not required but can generate faults.

Type: uint8\_t

Default value: 0

**walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0



## 3.109 ARMNeoverseV2CT

Defined in `LISA/ARMNeoverseV2CT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following parameters were added:

- `has_delayed_dbgreg`
- `has_delayed_sysreg`

### About ARMNeoverseV2CT

The model supports the following features:

- DynamIQ r3p0.
- DynamIQ(TM) Shared Unit-110 (DSU-110) system registers.
- L2 cache is supported at the per-core level only and there is no implementation of Core-Complex with shared L2 cache yet.
- PChannel for the cluster and for each core.
- `BROADCASTPERSIST` pin.
- Optional peripheral port.
- L3Cache partition.
- Per-core clock.
- Utility bus.

Support for the following features is planned for a future release:

- `BROADCASTCACHEMAINTPOU` pin.
- `COREINSTRRET`, `COREINSTRRUN`, and `nPMBIRQ` signals.
- Core-Complex.
- Each thread currently has its own L1Cache and L2Cache instead of a common cache that is shared by all threads of the core.
- Per-thread parameters are not yet supported but signals are implemented.

The following features will not be implemented:

- DynamIQ features that are irrelevant to the programmers' view simulation, for example:
  - Automatic CPU retention mode.

- Level-3 Cache RAM retention.
- 256-bit wide output transactions.
- Error correction or detection features.
- Self-test features (MBIST).
- Snoop filtering.
- Latency configuration
- Cache stashing capability
- Embedded Logic Analyzer (ELA).

This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (`crypto.dll` or `crypto.so`) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

### Iris and MTI instances for ARMNeoverseV2CT

This model has the following Iris instances:

Name	Instance type
ARMNeoverseV2CT	Cluster_ARM_Neoverse-V2
ARMNeoverseV2CT.AMU	PVBusLogger
ARMNeoverseV2CT.AMU.mapper	PVBusMapper
ARMNeoverseV2CT.DAP	PVBusLogger
ARMNeoverseV2CT.DAP.mapper	PVBusMapper
ARMNeoverseV2CT.DSU	DSU-110
ARMNeoverseV2CT.DSU.PPU_cluster	PPUv1
ARMNeoverseV2CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMNeoverseV2CT.DSU.PPU_core0	PPUv1
ARMNeoverseV2CT.DSU.PPU_core0.busslave	PVBusSlave
ARMNeoverseV2CT.DSU.mpam_busslave	PVBusSlave
ARMNeoverseV2CT.DSU.shared_cache	PVCache
ARMNeoverseV2CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseV2CT.DSU.shared_cache.upstream[Y] (where Y = 0-3)	PVBusSlave
ARMNeoverseV2CT.DSU.utility_slave[0]	PVBusSlave
ARMNeoverseV2CT.MMAP	PVBusLogger
ARMNeoverseV2CT.MMAP.mapper	PVBusMapper
ARMNeoverseV2CT.RAS	PVBusLogger
ARMNeoverseV2CT.RAS.mapper	PVBusMapper
ARMNeoverseV2CT.cpu0	ARM_Neoverse-V2
ARMNeoverseV2CT.cpu0.UTLB	TLB
ARMNeoverseV2CT.cpu0.debug_rom	debug_rom
ARMNeoverseV2CT.cpu0.dtlb	TLB
ARMNeoverseV2CT.cpu0.l1dcache	PVCache

Name	Instance type
ARMNeoverseV2CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseV2CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMNeoverseV2CT.cpu0.l1licache	PVCache
ARMNeoverseV2CT.cpu0.l1licache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseV2CT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMNeoverseV2CT.cpu0.l2cache	PVCache
ARMNeoverseV2CT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseV2CT.cpu0.l2cache.upstream[U] (where U = 0-1)	PVBusSlave
ARMNeoverseV2CT.default_CTM	EmbeddedCTM
ARMNeoverseV2CT.ext_bus	PVBusLogger
ARMNeoverseV2CT.ext_bus.mapper	PVBusMapper
ARMNeoverseV2CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMNeoverseV2CT.global_debug_rom	debug_rom
ARMNeoverseV2CT.secondary_debug_rom	debug_rom
ARMNeoverseV2CT.sve	ScalableVectorExtension

This model has the following MTI trace components:

Name	Component type
ARMNeoverseV2CT.AMU	PVBusLogger
ARMNeoverseV2CT.AMU.mapper	PVBusMapper
ARMNeoverseV2CT.DAP	PVBusLogger
ARMNeoverseV2CT.DAP.mapper	PVBusMapper
ARMNeoverseV2CT.DSU	DSU-110
ARMNeoverseV2CT.DSU.PPU_cluster	PPUv1
ARMNeoverseV2CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMNeoverseV2CT.DSU.PPU_core0	PPUv1
ARMNeoverseV2CT.DSU.PPU_core0.busslave	PVBusSlave
ARMNeoverseV2CT.DSU.mpam_busslave	PVBusSlave
ARMNeoverseV2CT.DSU.shared_cache	PVCache
ARMNeoverseV2CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseV2CT.DSU.shared_cache.upstream[Y] (where Y = 0-3)	PVBusSlave
ARMNeoverseV2CT.DSU.utility_slave[0]	PVBusSlave
ARMNeoverseV2CT.MMAP	PVBusLogger
ARMNeoverseV2CT.MMAP.mapper	PVBusMapper
ARMNeoverseV2CT.RAS	PVBusLogger
ARMNeoverseV2CT.RAS.mapper	PVBusMapper
ARMNeoverseV2CT.cpu0	ARM_Neoverse-V2
ARMNeoverseV2CT.cpu0.UTLB	TLB
ARMNeoverseV2CT.cpu0.l1dcache	PVCache
ARMNeoverseV2CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster

Name	Component type
ARMNeoverseV2CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMNeoverseV2CT.cpu0.l1icache	PVCache
ARMNeoverseV2CT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseV2CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMNeoverseV2CT.cpu0.l2cache	PVCache
ARMNeoverseV2CT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseV2CT.cpu0.l2cache.upstream[U] (where U = 0-1)	PVBusSlave
ARMNeoverseV2CT.ext_bus	PVBusLogger
ARMNeoverseV2CT.ext_bus.mapper	PVBusMapper
ARMNeoverseV2CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

### Ports for ARMNeoverseV2CT

Port	Direction	Protocol	Description
acp_s	slave	PVBus	AXI ACP slave port.
AEND0MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND2MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND3MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART2MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	slave	Signal	CHI defined pins.
broadcastcachemaint	slave	Signal	ACE defined pins.
broadcastouter	slave	Signal	ACE defined pins.
broadcastpersist	slave	Signal	CHI defined pins.
cfgend	slave	Signal	This signal is for EE bit initialisation.
cfgsdisable	slave	Signal	This signal disables write access to some secure Interrupt Controller registers.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcsn_pchannel	master	PChannel	Cluster PCSM signal

Port	Direction	Protocol	Description
cluster_powerdown_out	master	Signal	DEPRECATED - An output from PPU that informs thermal controller of the cluster power info
cluster_ppu_hw_stat	master	Value	Cluster PPU power state
clusterid	slave	Value	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set ManagerID64 of the core.
clusterpmuirq	master	Signal	DynamiQ pmu irq
CNTHPIRQ	master	Signal	Timer signals to SOC.
CNTHPSIRQ	master	Signal	Timer signals to SOC
CNTHVIRQ	master	Signal	Timer signals to SOC.
CNTHVSIRQ	master	Signal	Timer signals to SOC
CNTPNSIRQ	master	Signal	Timer signals to SOC.
CNTPSIRQ	master	Signal	Timer signals to SOC.
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.
CNTVIRQ	master	Signal	Timer signals to SOC.
commirq	master	Signal	Interrupt signal from debug communications channel.
core_clk_in	slave	ClockSignal	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
core_pcsn_pchannel	master	PChannel	Core PCSM signals
core_powerdown_out	master	Signal	DEPRECATED - An output from PPU that informs thermal controller of the core power info
core_ppu_hw_stat	master	Value	Notify the power state change inside the PPU.
coreerrirq	master	Signal	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq	master	Signal	Fault indicator for a detected 1 or 2 bit ECC error
coreinstrrun	master	Signal	Core Running State
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
ctidbgirq	master	Signal	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	slave	Signal	External debug interface.
dbgnopwrdown	master	Signal	No power-down request.
dbgprupreq	master	Signal	Debug power up request.
defaultMP	slave	Signal	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main master interface ports.
dev_debug_s	slave	PVBus	External debug interface.

Port	Direction	Protocol	Description
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.
gicreset	master	Signal	An output from PPU that can be used to reset GICCLK domain components
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
memorymapped_debug_s	slave	PVBus	External debug interface.
pmbirq	master	Signal	Interrupt signal from the statistical profiling unit.
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	master	Signal	PPU cluster interrupt.
ppu_cluster_isolate	master	Signal	PPU Cluster Domain isolation control.
ppu_cluster_wakerequest	slave	Signal	PPU cluster wake request signal.
ppu_core_irq	master	Signal	PPU core interrupt.
ppu_core_isolate	master	Signal	PPU Core Domain isolation control.
ppu_core_wakerequest	slave	Signal	PPU Core wake request signals.
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	master	PVBus	The core will generate bus requests on this port.
pvbus_periph_m	master	PVBus	The core can generate peripheral bus request on this port.
rei	slave	Signal	Per core RAM Error Interrupt.
reset	slave	Signal	Reset signal to cluster.
rvbaraddr	slave	Value_64	Reset vector base address.
sci_m	master	SystemCoherencyInterface	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei	slave	Signal	Per core System Error physical pins.
spiden	slave	Signal	External debug interface.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq	master	Signal	Interrupt signal from the trace buffer unit.
utility_bus_s	slave	PVBus	Utility bus slave
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfiq	slave	Signal	Virtualised FIQ.
virq	slave	Signal	Virtualised IRQ.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Per core virtual System Error physical pins.

## Parameters for ARMNeoverseV2CT

### **cpu0.CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **cpu0.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **cpu0.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

### **cpu0.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

### **cpu0.force-fpsid**

Override the FPSID value.

Type: `bool`

Default value: `true`

### **cpu0.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**cpu0.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**cpu0.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**cpu0.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**cpu0.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**cpu0.l2cache-size**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x100000`

**cpu0.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`



Default value: 0x0

### **cpu0.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **cpu0.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **cpu0.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **cpu0.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

### **cpu0.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

### **cpu0.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**cpu0.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

**cpu0.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**cpu0.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: uint8\_t

Default value: 60

**cpu0.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: uint8\_t

Default value: 171

**cpu0.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: string

Default value: N/A

**cpu0.semihosting-cwd**

Base directory for semihosting file access.

Type: string

Default value: N/A

**cpu0.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: bool

Default value: `true`

**`cpu0.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

**`cpu0.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

**`cpu0.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**`cpu0.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**`cpu0.trace_special_hlt_imm16`**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**`cpu0.vfp-enable_at_reset`**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**cpu0.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**AEND0\_DEFAULT**

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

Type: `uint64_t`

Default value: `0x0`

**AEND1\_DEFAULT**

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

Type: `uint64_t`

Default value: `0x0`

**AEND2\_DEFAULT**

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

Type: `uint64_t`

Default value: `0x0`

**AEND3\_DEFAULT**

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART0\_DEFAULT**

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART1\_DEFAULT**

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART2\_DEFAULT**

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART3\_DEFAULT**

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

Type: `uint64_t`

Default value: `0x0`

**BROADCASTATOMIC**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`

Default value: `true`

**BROADCASTCACHEMAINT**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

Type: `bool`

Default value: `false`

**BROADCASTOUTER**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

Type: `bool`

Default value: `false`

**BROADCASTPERSIST**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: `bool`

Default value: `true`

**CLUSTER\_ID**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the ManagerID64 of the core.

Type: `uint32_t`

Default value: `0x0`

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. This value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required.

Type: `uint8_t`

Default value: `1`

**CPUCFR**

Value of CPU Configuration Register.

Type: `uint64_t`

Default value: `0x0`

**GICDISABLE**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`

Default value: `true`

**NUM\_CORES**

Number of cores per cluster.

Type: `uint8_t`

Default value: 1

**bus\_type**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

Type: `uint8_t`

Default value: 0

**core\_power\_on\_by\_default**

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPUs which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

Type: `bool`

Default value: `false`

**cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-maintenance\_latency**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-miss\_latency**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-prefetch\_enabled**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

Type: `bool`

Default value: `false`

**dcache-read\_access\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-size**

L1 D-Cache size in bytes.

Type: `uint32_t`



Default value: 0x8000

### **dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **dcache-state\_modelled**

Set whether D-cache has stateful implementation.

Type: bool

Default value: false

### **dcache-write\_access\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **default\_opmode**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

Type: uint8\_t

Default value: 4

### **diagnostics**

Enable DynamIQ diagnostic messages.

Type: bool

Default value: `false`

### **`enable_simulation_performance_optimizations`**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of `stage12_tlb_size` parameter to 1024).

Type: `bool`

Default value: `true`

### **`ete.CLAIMTAGS`**

Number of claim tags.

Type: `uint8_t`

Default value: 32

### **`ete.MAX_INST_PER_Q`**

Maximum limit for the number of instructions implied by a Q element.

Type: `uint16_t`

Default value: 0x1

### **`ete.NumberOfRSPairs`**

Number of resource selector pairs.

Type: `uint8_t`

Default value: 8

### **`ete.PIDR_CMOD`**

TRCPIDR CMOD value.

Type: `uint8_t`

Default value: 0

### **`ete.PIDR_REVAND`**

TRCPIDR REVAND value.

Type: `uint8_t`

Default value: 0

**ete.PIDR\_REVISION**

TRCPIDR REVISION value.

Type: `uint8_t`

Default value: 0

**ete.Q\_CADENCE**

Number of instruction blocks traced between two Q elements.

Type: `uint16_t`

Default value: 0x1

**ete.RES0\_STATEFUL**

Whether **RES0** bits are stateful or **RAZ/WI**.

Type: `bool`

Default value: `false`

**ete.RETSTACK**

Return stack depth.

Type: `uint8_t`

Default value: 3

**ete.REVISION**

TRCIDR1 revision value.

Type: `uint8_t`

Default value: 0

**ete.SIM\_OVERFLOW\_GRANULARITY**

Number of instruction blocks in each granule, for simulated overflow.

Type: `uint32_t`

Default value: 0x64

**ete.SIM\_OVERFLOW\_PERCENTAGE**

Percentage of instruction blocks lost in each granule, for simulated overflow.

Type: `uint8_t`

Default value: 0

**ete.SOURCE\_ADDRESS**

Allow generation of source address elements.

Type: `bool`

Default value: `false`

**ete.TRACE\_OUTPUT**

File to which to write trace byte stream.

Type: `string`

Default value: `N/A`

**ete.TRCRSRTA\_FORCED\_EXCEP**

TRCRSR.TA value for a forcibly traced exception.

Type: `bool`

Default value: `false`

**ext\_abort\_normal\_noncacheable\_read\_is\_sync**

Synchronous reporting of normal noncacheable-read external aborts.

Type: `bool`

Default value: `true`

**ext\_abort\_so\_write\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `uint8_t`

Default value: 2

**force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks**

Force MTE tag accesses to **RAZ/WI** and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

Type: `bool`

Default value: `false`

**force\_zero\_PSTATE\_PAN**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

**force\_zero\_mpam\_partid\_and\_pmg**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAM0\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

Type: `uint8_t`

Default value: `0`

**has\_coherent\_icache**

Whether icache invalidation to the point of unification is required for instruction to data coherence.  
`true` - Invalidate operations not required.

Type: `bool`

Default value: `true`

**has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

**has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

**has\_enhanced\_pan**

Implements Armv8.7 Enhanced PAN feature (FEAT\_PAN3) values of this parameter are:- 1, feature is implemented if ARMv8.7 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: `2`

**has\_ete**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (-plugin or -P).

Type: `bool`

Default value: `false`

**has\_external\_rndr**

Implement external random number generator module. When enabling this with `has_rndr` enabled, the external random number generator will be used instead of internal random number generator.

Type: `uint8_t`

Default value: `1`

**has\_large\_va**

Implement support for the extended 52-bit virtual addresses from ARMv8.2 (FEAT\_LVA). values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.2 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: `0`

**has\_peripheral\_port**

If true, an additional AXI peripheral port is configured.

Type: `bool`

Default value: `false`

**has\_rndr**

Implement random number instructions to read from RNDR and RNDRSS random number registers from ARMv8.5 (FEAT\_RNG).

Type: `uint8_t`

Default value: `1`

**icache-hit\_latency**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

**icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-size**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: `0x8000`

### **`icache-state_modelled`**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **`instruction_tlb_size`**

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

Type: `uint32_t`

Default value: `0x0`

### **`invalidate_code_cache_on_icache_cmo`**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: `uint8_t`

Default value: `0`

### **`memory_tagging_support_level`**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

Type: `uint8_t`

Default value: `3`

### **`mpam_max_partid`**

MPAM Maximum PARTID Supported.

Type: `uint16_t`

Default value: `511`

### **`mpam_max_vpmr`**

MPAM Maximum VPMR Supported.

Type: `uint8_t`

Default value: `7`



**mpmm\_accumulator\_multiplier**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator.value of n means the accumulator will use (n \* accumulator value) to calculate the mpmm threshold (MPMM).is provided as a fast model workaround to handle cases where execution of quantum in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

Type: uint8\_t

Default value: 1

**pmu-num\_counters**

Number of PMU counters implemented.

Type: uint8\_t

Default value: 6

**ptw\_latency**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

**rndr\_rndrrs\_seed**

Initial seed for random engine used in RNDR register.

Type: uint64\_t

Default value: 0x0

**stage12\_tlb\_size**

Number of stage1+2 tlb entries. Valid values are 0 or >= 4.

Type: uint32\_t

Default value: 0x80

**tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

**tlbi\_stall\_enabled**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: `bool`

Default value: `false`

**trace\_physical\_registers\_when\_host\_virtualisation\_enabled**

Trace sysreg accesses to physical register accessed (0=disabled, 1=Trace only ELR/SPSR\_EL1 as ELR/SPSR\_EL2, 2=Trace all redirected registers as physical registers affecting all features inc. the host virtualised registers.

Type: `uint8_t`

Default value: `1`

**treat\_PAC\_as\_NOP**

Non-architecture parameter. Treat Pointer Authentication as **NOP**. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are **NOP**. Branch, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

Type: `bool`

Default value: `false`

**walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

## 3.110 ARMNeoverseV3AECT

Defined in `LISA/ARMNeoverseV3AECT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

## Changes in 11.31.15

The following parameters were added:

- `has_delayed_dbgreg`
- `has_delayed_sysreg`
- `mpam_has_altsp`

The following parameters were removed:

- `num_acp`

## About ARMNeoverseV3AECT

A DSU-120 DynamIQ cluster containing a single Neoverse-V3AE core configured for Direct connect.

The core supports the following optional features:

- Realm Management Extension
- Coherent instruction cache

The following features are fully or partially turned off by default for performance reasons, but may be fully enabled by parameters:

- BRBE
- ETE

The following DSU/CPU features will not be implemented:

- DynamIQ features that are irrelevant to the programmers' view simulation.
- 256-bit wide output transactions.
- ECC and parity schemes are hardware-specific so are not supported.
- Self-test features (MBIST).
- Snoop filtering.
- Cache stashing capability.
- Embedded Logic Analyzer (ELA).

This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (`crypto.dll` or `crypto.so`) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

## Iris and MTI instances for ARMNeoverseV3AECT

This model has the following Iris instances:

Name	Instance type
ARMNeoverseV3AECT	Cluster_ARM_Neoverse-V3AE
ARMNeoverseV3AECT.AMU	PVBusLogger

Name	Instance type
ARMNeoverseV3AECT.AMU.mapper	PVBusMapper
ARMNeoverseV3AECT.DAP	PVBusLogger
ARMNeoverseV3AECT.DAP.mapper	PVBusMapper
ARMNeoverseV3AECT.DSU	DSU-120
ARMNeoverseV3AECT.DSU.PPU_cluster	PPUv1
ARMNeoverseV3AECT.DSU.PPU_cluster.busslave	PVBusSlave
ARMNeoverseV3AECT.DSU.PPU_core0	PPUv1
ARMNeoverseV3AECT.DSU.PPU_core0.busslave	PVBusSlave
ARMNeoverseV3AECT.DSU.mpam_busslave	PVBusSlave
ARMNeoverseV3AECT.DSU.shared_cache	PVCache
ARMNeoverseV3AECT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseV3AECT.DSU.shared_cache.upstream[Y] (where Y = 0–4)	PVBusSlave
ARMNeoverseV3AECT.DSU.utility_slave[0]	PVBusSlave
ARMNeoverseV3AECT.MMAP	PVBusLogger
ARMNeoverseV3AECT.MMAP.mapper	PVBusMapper
ARMNeoverseV3AECT.RAS	PVBusLogger
ARMNeoverseV3AECT.RAS.mapper	PVBusMapper
ARMNeoverseV3AECT.cpu0	ARM_Neoverse-V3AE
ARMNeoverseV3AECT.cpu0.UTLB	TLB
ARMNeoverseV3AECT.cpu0.debug_rom	debug_rom
ARMNeoverseV3AECT.cpu0.dtlb	TLB
ARMNeoverseV3AECT.cpu0.l1dcache	PVCache
ARMNeoverseV3AECT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseV3AECT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMNeoverseV3AECT.cpu0.l1licache	PVCache
ARMNeoverseV3AECT.cpu0.l1licache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseV3AECT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMNeoverseV3AECT.cpu0.l2cache	PVCache
ARMNeoverseV3AECT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseV3AECT.cpu0.l2cache.upstream[U] (where U = 0–1)	PVBusSlave
ARMNeoverseV3AECT.default_CTM	EmbeddedCT
ARMNeoverseV3AECT.ext_bus	PVBusLogger
ARMNeoverseV3AECT.ext_bus.mapper	PVBusMapper
ARMNeoverseV3AECT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMNeoverseV3AECT.global_debug_rom	debug_rom
ARMNeoverseV3AECT.secondary_debug_rom	debug_rom
ARMNeoverseV3AECT.sve	ScalableVectorExtension

This model has the following MTI trace components:

Name	Component type
ARMNeoverseV3AECT.AMU	PVBusLogger
ARMNeoverseV3AECT.AMU.mapper	PVBusMapper
ARMNeoverseV3AECT.DAP	PVBusLogger
ARMNeoverseV3AECT.DAP.mapper	PVBusMapper
ARMNeoverseV3AECT.DSU	DSU-120
ARMNeoverseV3AECT.DSU.PPU_cluster	PPUv1
ARMNeoverseV3AECT.DSU.PPU_cluster.busslave	PVBusSlave
ARMNeoverseV3AECT.DSU.PPU_core0	PPUv1
ARMNeoverseV3AECT.DSU.PPU_core0.busslave	PVBusSlave
ARMNeoverseV3AECT.DSU.mpam_busslave	PVBusSlave
ARMNeoverseV3AECT.DSU.shared_cache	PVCache
ARMNeoverseV3AECT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseV3AECT.DSU.shared_cache.upstream[Y] (where Y = 0-4)	PVBusSlave
ARMNeoverseV3AECT.DSU.utility_slave[0]	PVBusSlave
ARMNeoverseV3AECT.MMAP	PVBusLogger
ARMNeoverseV3AECT.MMAP.mapper	PVBusMapper
ARMNeoverseV3AECT.RAS	PVBusLogger
ARMNeoverseV3AECT.RAS.mapper	PVBusMapper
ARMNeoverseV3AECT.cpu0	ARM_Neoverse-V3AE
ARMNeoverseV3AECT.cpu0.UTLB	TLB
ARMNeoverseV3AECT.cpu0.l1dcache	PVCache
ARMNeoverseV3AECT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseV3AECT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMNeoverseV3AECT.cpu0.l1icache	PVCache
ARMNeoverseV3AECT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseV3AECT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMNeoverseV3AECT.cpu0.l2cache	PVCache
ARMNeoverseV3AECT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseV3AECT.cpu0.l2cache.upstream[U] (where U = 0-1)	PVBusSlave
ARMNeoverseV3AECT.ext_bus	PVBusLogger
ARMNeoverseV3AECT.ext_bus.mapper	PVBusMapper
ARMNeoverseV3AECT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

## Ports for ARMNeoverseV3AECT

Port	Direction	Protocol	Description
acp_s	slave	PVBus	AXI ACP slave port.
AEND0MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).

Port	Direction	Protocol	Description
AEND2MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND3MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART2MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	slave	Signal	CHI defined pins.
broadcastcachemaint	slave	Signal	ACE defined pins.
broadcastouter	slave	Signal	ACE defined pins.
broadcastpersist	slave	Signal	CHI defined pins.
cfgend	slave	Signal	This signal is for EE bit initialisation.
cfgsdisable	slave	Signal	This signal disables write access to some secure Interrupt Controller registers.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcsmp_channel	master	PChannel	Cluster PCSM signal
cluster_powerdown_out	master	Signal	DEPRECATED - An output from PPU that informs thermal controller of the cluster power info
cluster_ppu_hw_stat	master	Value	Cluster PPU power state
clusterid	slave	Value	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set ManagerID64 of the core.
clusterpmuirq	master	Signal	DynamiQ pmu irq
CNTHPIRQ	master	Signal	Timer signals to SOC.
CNTHPSIRQ	master	Signal	Timer signals to SOC
CNTHVIRQ	master	Signal	Timer signals to SOC.
CNTHVSIRQ	master	Signal	Timer signals to SOC
CNTPNSIRQ	master	Signal	Timer signals to SOC.
CNTPSIRQ	master	Signal	Timer signals to SOC.
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.

Port	Direction	Protocol	Description
CNTVIRQ	master	Signal	Timer signals to SOC.
commirq	master	Signal	Interrupt signal from debug communications channel.
core_clk_in	slave	ClockSignal	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
core_pcsn_pchannel	master	PChannel	Core PCSM signals
core_powerdown_out	master	Signal	DEPRECATED - An output from PPUs that informs thermal controller of the core power info
core_ppu_hw_stat	master	Value	Notify the power state change inside the PPU.
coreerrirq	master	Signal	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq	master	Signal	Fault indicator for a detected 1 or 2 bit ECC error
coreinstrrun	master	Signal	Core Running State
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
ctidbgirq	master	Signal	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	slave	Signal	External debug interface.
dbgnopwrdown	master	Signal	No power-down request.
dbgprupreq	master	Signal	Debug power up request.
defaultMP	slave	Signal	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main master interface ports.
dev_debug_s	slave	PVBus	External debug interface.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.
gicreset	master	Signal	An output from PPUs that can be used to reset GICCLK domain components
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
l0gptsz	slave	Value	RME LOGPTSZ port
legacy_tz_en	slave	Signal	RME LEGACY_TZ_EN port
memorymapped_debug_s	slave	PVBus	External debug interface.
pmbirq	master	Signal	Interrupt signal from the statistical profiling unit.
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	master	Signal	PPU cluster interrupt.
ppu_cluster_isolate	master	Signal	PPU Cluster Domain isolation control.
ppu_cluster_wakerequest	slave	Signal	PPU cluster wake request signal.
ppu_core_irq	master	Signal	PPU core interrupt.
ppu_core_isolate	master	Signal	PPU Core Domain isolation control.
ppu_core_wakerequest	slave	Signal	PPU core wake request signal.

Port	Direction	Protocol	Description
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbush_m0	master	PVBus	The core will generate bus requests on this port.
pvbush_periph_m	master	PVBus	The core can generate peripheral bus request on this port.
rei	slave	Signal	Per core RAM Error Interrupt.
reset	slave	Signal	Reset signal to cluster.
rlpiden	slave	Signal	External debug interface.
rtpiden	slave	Signal	External debug interface.
rvbaraddr	slave	Value_64	Reset vector base address.
sci_m	master	SystemCoherencyInterface	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain
sei	slave	Signal	Per core System Error physical pins.
spiden	slave	Signal	External debug interface.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq	master	Signal	Interrupt signal from the trace buffer unit.
utility_bus_s	slave	PVBus	Utility bus slave
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfiq	slave	Signal	Virtualised FIQ.
virq	slave	Signal	Virtualised IRQ.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Per core virtual System Error physical pins.

## Parameters for ARMNeoverseV3AECT

### cpu0 . CFGEND

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### cpu0 . CFGTE

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`



**cpu0.CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

**cpu0.RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**cpu0.crypto\_aes**

AES instructions supported (requires CryptoPlugin to be loaded). 0, not implemented. 2, AES and PMULL instructions implemented (FEAT\_AES, FEAT\_PMULL).

Type: `uint8_t`

Default value: 2

**cpu0.crypto\_sha3**

Implement ARMv8.4 SHA-3 instructions (requires CryptoPlugin to be loaded) (FEAT\_SHA3).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 2

**cpu0.crypto\_sha512**

Implement ARMv8.4 SHA-512 instructions (requires CryptoPlugin to be loaded) (FEAT\_SHA512).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 2

**cpu0.crypto\_sm3**

Implement ARMv8.4 SM-3 instructions (requires CryptoPlugin to be loaded) (FEAT\_SM3).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 2

**cpu0.crypto\_sm4**

Implement ARMv8.4 SM-4 instructions (requires CryptoPlugin to be loaded) (FEAT\_SM4).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 2

**cpu0.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter `trace_special_hlt_imm16`.

Type: `bool`

Default value: `false`

**cpu0.l2cache-hit\_latency**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**cpu0.l2cache-maintenance\_latency**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**cpu0.l2cache-miss\_latency**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**cpu0.l2cache-read\_access\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **cpu0.l2cache-read\_latency**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. l2cache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **cpu0.l2cache-size**

L2 Cache size in bytes.

Type: uint32\_t

Default value: 0x80000

### **cpu0.l2cache-snoop\_data\_transfer\_latency**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **cpu0.l2cache-snoop\_issue\_latency**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **cpu0.l2cache-ways**

L2 Cache number of ways (sets are implicit from size).

Type: uint8\_t

Default value: 8

### **cpu0.l2cache-write\_access\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if l2cache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **cpu0.l2cache-write\_latency**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. l2cache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **cpu0.max\_code\_cache\_mb**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: uint16\_t

Default value: 0x100

### **cpu0.min\_sync\_level**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: uint8\_t

Default value: 0

### **cpu0.semihosting-A32\_HLT**

A32 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

### **cpu0.semihosting-A64\_HLT**

A64 HLT number for semihosting calls.

Type: uint16\_t

Default value: 0xf000

### **cpu0.semihosting-ARM\_SVC**

A32 SVC number for semihosting calls.

Type: uint32\_t

Default value: 0x123456

**cpu0.semihosting-T32\_HLT**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: 60

**cpu0.semihosting-Thumb\_SVC**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: 171

**cpu0.semihosting-cmd\_line**

Command line available to semihosting calls.

Type: `string`

Default value: N/A

**cpu0.semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

**cpu0.semihosting-enable**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

**cpu0.semihosting-heap\_base**

Virtual address of heap base.

Type: `uint64_t`

Default value: 0x0

**cpu0.semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint64_t`

Default value: 0xf000000

**cpu0.semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

**cpu0.semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**cpu0.trace\_special\_hlt\_imm16**

For this HLT number, IF `enable_trace_special_hlt_imm16=true`, skip performing usual HLT execution but call MTI trace if registered.

Type: `uint16_t`

Default value: `0xf000`

**cpu0.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: `bool`

Default value: `false`

**cpu0.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `true`

**AEND0\_DEFAULT**

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMPO input signal).

Type: `uint64_t`

Default value: `0x0`

**AEND1\_DEFAULT**

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

Type: `uint64_t`

Default value: `0x0`

**AEND2\_DEFAULT**

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

Type: `uint64_t`

Default value: `0x0`

**AEND3\_DEFAULT**

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART0\_DEFAULT**

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART1\_DEFAULT**

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART2\_DEFAULT**

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART3\_DEFAULT**

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

Type: `uint64_t`

Default value: `0x0`

**BROADCASTATOMIC**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`

Default value: `true`

**BROADCASTCACHEMAINT**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

Type: `bool`

Default value: `false`

**BROADCASTOUTER**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

Type: `bool`

Default value: `false`

**BROADCASTPERSIST**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: `bool`

Default value: `true`

**CLUSTER\_ID**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the ManagerID64 of the core.



Type: `uint32_t`

Default value: `0x0`

### **CMO\_broadcast\_when\_cache\_state\_modelling\_disabled**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. This value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain even when cache state modelling is disabled. Possible values are: - 0 = All CMOs are broadcast if architecturally required - 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required.

Type: `uint8_t`

Default value: `1`

### **CPUCFR**

Value of CPU Configuration Register.

Type: `uint64_t`

Default value: `0x20`

### **GICDISABLE**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`

Default value: `true`

### **NUM\_CORES**

Number of cores per cluster.

Type: `uint8_t`

Default value: `1`

### **brbe\_disable\_recording**

If BRBE is implemented and this is set to true, disable BRBE recording. All registers will be functional, but no branches will be recorded. This will improve model performance for workloads that enable BRBE, but don't care about the information stored in it. (FEAT\_BRBE).

Type: `bool`

Default value: `true`

**brbe\_log2\_num\_records**

Log2 of number of BRB records supported. 3 -> 8 records, ... 6 -> 64 records.

Type: `uint8_t`

Default value: 5

**bus\_type**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

Type: `uint8_t`

Default value: 0

**core\_power\_on\_by\_default**

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPUs which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

Type: `bool`

Default value: `false`

**cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-maintenance\_latency**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-miss\_latency**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-prefetch\_enabled**

Enable simulation of data cache prefetching. This is only used when `dcache-state_modelled=true`.

Type: `bool`

Default value: `false`

**dcache-read\_access\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size `dcache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. `dcache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-size**

L1 D-Cache size in bytes.

Type: `uint32_t`

Default value: 0x10000

### **dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **dcache-state\_modelled**

Set whether D-cache has stateful implementation.

Type: bool

Default value: false

### **dcache-write\_access\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size dcache-write\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-write\_latency is set. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **dcache-write\_latency**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. dcache-write\_access\_latency must be set to 0 for per-byte latencies to be applied. This is only used when dcache-state\_modelled=true.

Type: uint64\_t

Default value: 0x0

### **default\_opmode**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

Type: uint8\_t

Default value: 4

### **diagnostics**

Enable DynamIQ diagnostic messages.

Type: bool

Default value: `false`

### **ecv\_support\_level**

Implement Enhanced Counter Virtualization feature from ARMv8.6. 0, Not supported. 1, fully supported without CNTPOFF. 2, fully supported with CNTPOFF (FEAT\_ECV).

Type: `uint8_t`

Default value: 2

### **enable\_simulation\_performance\_optimizations**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of `stage12_tlb_size` parameter to 1024).

Type: `bool`

Default value: `true`

### **ete.CLAIMTAGS**

Number of claim tags.

Type: `uint8_t`

Default value: 4

### **ete.MAX\_INST\_PER\_Q**

Maximum limit for the number of instructions implied by a Q element.

Type: `uint16_t`

Default value: `0x1`

### **ete.NumberOfRSPairs**

Number of resource selector pairs.

Type: `uint8_t`

Default value: 8

### **ete.PIDR\_CM0D**

TRCPIDR CM0D value.

Type: `uint8_t`

Default value: 0

**ete.Q\_CADENCE**

Number of instruction blocks traced between two Q elements.

Type: `uint16_t`

Default value: `0x1`

**ete.RES0\_STATEFUL**

Whether **RES0** bits are stateful or **RAZ/WI**.

Type: `bool`

Default value: `false`

**ete.RETSTACK**

Return stack depth.

Type: `uint8_t`

Default value: `3`

**ete.SIM\_OVERFLOW\_GRANULARITY**

Number of instruction blocks in each granule, for simulated overflow.

Type: `uint32_t`

Default value: `0x64`

**ete.SIM\_OVERFLOW\_PERCENTAGE**

Percentage of instruction blocks lost in each granule, for simulated overflow.

Type: `uint8_t`

Default value: `0`

**ete.SOURCE\_ADDRESS**

Allow generation of source address elements.

Type: `bool`

Default value: `false`

**ete.TRACE\_OUTPUT**

File to which to write trace byte stream.

Type: `string`

Default value: `N/A`

**ete.TRCRSRTA\_FORCED\_EXCEP**

TRCRSR.TA value for a forcibly traced exception.

Type: `bool`

Default value: `false`

**ext\_abort\_so\_write\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `uint8_t`

Default value: 2

**force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks**

Force MTE tag accesses to **RAZ/WI** and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

Type: `bool`

Default value: `false`

**force\_zero\_PSTATE\_PAN**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

**force\_zero\_mpam\_partid\_and\_pmg**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAMO\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

Type: `uint8_t`

Default value: 0

**has\_coherent\_icache**

Whether icache invalidation to the point of unification is required for instruction to data coherence. true - Invalidate operations not required.

Type: `bool`

Default value: `true`

### **has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

### **has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

### **has\_enhanced\_pan**

Implements Armv8.7 Enhanced PAN feature (FEAT\_PAN3) values of this parameter are:- 1, feature is implemented if ARMv8.7 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 2

### **has\_ete**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (`-plugin` or `-P`).

Type: `bool`

Default value: `false`

### **has\_peripheral\_port**

If true, an additional AXI peripheral port is configured.

Type: `bool`

Default value: `false`

### **has\_rndr**

Implement random number instructions to read from RNDR and RNDRSS random number registers from ARMv8.5 (FEAT\_RNG). values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.5 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1



**has\_statistical\_profiling**

Whether Statistical Based Profiling is implemented (FEAT\_SPE).

Type: `bool`

Default value: `true`

**has\_v8\_7\_spe\_inverted\_filtering**

Where FEAT\_SPEv1p2 is implemented, whether inverted filtering by events is implemented (represented by PMISDR.FnE).

Type: `bool`

Default value: `false`

**has\_v8\_7\_spe\_previous\_branch\_target**

Where FEAT\_SPEv1p2 is implemented, whether the optional branch target feature is implemented (FEAT\_SPE\_PBT).

Type: `bool`

Default value: `false`

**icache-hit\_latency**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when icache-state\_modelled=true.

Type: `bool`

Default value: `false`

**icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size icache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if icache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when icache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. icache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when icache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**icache-size**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: `0x10000`

**icache-state\_modelled**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

**instruction\_tlb\_size**

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

Type: `uint32_t`

Default value: 0x0

### **invalidate\_code\_cache\_on\_icache\_cmo**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: `uint8_t`

Default value: 0

### **l3cache-ways**

L3 Cache number of ways (sets are implicit from size).

Type: `uint16_t`

Default value: 16

### **log2\_trace\_buffer\_alignment**

Log2 of trace buffer alignment constraint for output buffer (0->1B ... 11->2Kib).

Type: `uint8_t`

Default value: 6

### **memory\_tagging\_support\_level**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

Type: `uint8_t`

Default value: 3

### **mpam\_has\_altsp**

DEPRECATED: MPAMIDR\_EL1.HAS\_ALTSP is required when FEAT\_RME is implemented.

Type: `bool`

Default value: `false`

### **mpamidr\_has\_force\_ns**

Whether MPAMIDR\_EL1.HAS\_FORCE\_NS bit is set or clear.values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.6 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

**mpamidr\_has\_sdeflt**

Whether MPAMIDR\_EL1.HAS\_SDEFLT bit is set or clear. values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.6 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

**mpamidr\_has\_tidr**

Whether MPAMIDR\_EL1.HAS\_TIDR bit is set or clear. values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.6 is enabled.- 2, feature is implemented.

Type: uint8\_t

Default value: 1

**mpmm\_accumulator\_multiplier**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator. value of n means the accumulator will use (n \* accumulator value) to calculate the mpmm threshold (MPMM). is provided as a fast model workaround to handle cases where execution of quantum in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

Type: uint8\_t

Default value: 1

**ptw\_latency**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

**reported\_patch\_level**

Purely cosmetic patch level value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: int8\_t

Default value: -1

**reported\_revision\_number**

Purely cosmetic revision number value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: int8\_t

Default value: -1

**rme\_level0\_gpt\_size**

The range of address space protected by each entry in the level 0 GPT (0->1GB 1->16GB, 2->64GB, 3->512GB).

Type: `uint8_t`

Default value: 0

**rme\_support\_level**

0 -> Realm management extension not implemented, 1 -> LEGACY\_TZ\_EN mode i.e. RME register fields are stateful but only supports secure/non-secure states, 2 -> Realm management extension fully implemented (FEAT\_RME).

Type: `uint8_t`

Default value: 2

**rndr\_rndrrs\_seed**

Initial seed for random engine used in RNDR register.

Type: `uint64_t`

Default value: 0x0

**stage12\_tlb\_size**

Number of stage1+2 tlb entries. Valid values are 0 or >= 4.

Type: `uint32_t`

Default value: 0x80

**tcr\_txsz\_undersize\_should\_fault**

If large VA is not supported, Whether undersized TxSZ value should generate translation fault.

Type: `bool`

Default value: `false`

**tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: 0x0

**tlbi\_stall\_enabled**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: `bool`

Default value: `false`

**treat\_PAC\_as\_NOP**

Non-architecture parameter. Treat Pointer Authentication as **NOP**. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are **NOP**. Branch, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

Type: `bool`

Default value: `false`

**walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

## 3.111 ARMNeoverseV3CT

Defined in `LISA/ARMNeoverseV3CT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following parameters were added:

- `has_delayed_dbgreg`
- `has_delayed_sysreg`
- `mpam_has_altsp`

The following parameters were removed:

- `num_acp`

## About ARMNeoverseV3CT

A DSU-120 DynamIQ cluster containing a single Neoverse-V3 core configured for Direct connect.

The core supports the following optional features:

- Realm Management Extension
- Coherent instruction cache

The following features are fully or partially turned off by default for performance reasons, but may be fully enabled by parameters:

- BRBE
- ETE

The following DSU/CPU features will not be implemented:

- DynamIQ features that are irrelevant to the programmers' view simulation.
- 256-bit wide output transactions.
- ECC and parity schemes are hardware-specific so are not supported.
- Self-test features (MBIST).
- Snoop filtering.
- Cache stashing capability.
- Embedded Logic Analyzer (ELA).

This model supports the Armv8-A Cryptographic Extensions, which requires the crypto plug-in (`crypto.dll` or `crypto.so`) to be loaded. The crypto plug-in is available for download from the [Arm Developer website](#).

## Iris and MTI instances for ARMNeoverseV3CT

This model has the following Iris instances:

Name	Instance type
ARMNeoverseV3CT	Cluster_ARM_Neoverse-V3
ARMNeoverseV3CT.AMU	PVBusLogger
ARMNeoverseV3CT.AMU.mapper	PVBusMapper
ARMNeoverseV3CT.DAP	PVBusLogger
ARMNeoverseV3CT.DAP.mapper	PVBusMapper
ARMNeoverseV3CT.DSU	DSU-120
ARMNeoverseV3CT.DSU.PPU_cluster	PPUv1
ARMNeoverseV3CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMNeoverseV3CT.DSU.PPU_core0	PPUv1
ARMNeoverseV3CT.DSU.PPU_core0.busslave	PVBusSlave
ARMNeoverseV3CT.DSU.mpam_busslave	PVBusSlave
ARMNeoverseV3CT.DSU.shared_cache	PVCache

Name	Instance type
ARMNeoverseV3CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseV3CT.DSU.shared_cache.upstream[Y] (where Y = 0-4)	PVBusSlave
ARMNeoverseV3CT.DSU.utility_slave[0]	PVBusSlave
ARMNeoverseV3CT.MMAP	PVBusLogger
ARMNeoverseV3CT.MMAP.mapper	PVBusMapper
ARMNeoverseV3CT.RAS	PVBusLogger
ARMNeoverseV3CT.RAS.mapper	PVBusMapper
ARMNeoverseV3CT.cpu0	ARM_Neoverse-V3
ARMNeoverseV3CT.cpu0.UTLB	TLB
ARMNeoverseV3CT.cpu0.debug_rom	debug_rom
ARMNeoverseV3CT.cpu0.dtlb	TLB
ARMNeoverseV3CT.cpu0.l1dcache	PVCache
ARMNeoverseV3CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseV3CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMNeoverseV3CT.cpu0.l1icache	PVCache
ARMNeoverseV3CT.cpu0.l1icache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseV3CT.cpu0.l1icache.upstream[0]	PVBusSlave
ARMNeoverseV3CT.cpu0.l2cache	PVCache
ARMNeoverseV3CT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseV3CT.cpu0.l2cache.upstream[U] (where U = 0-1)	PVBusSlave
ARMNeoverseV3CT.default_CTM	EmbeddedCT
ARMNeoverseV3CT.ext_bus	PVBusLogger
ARMNeoverseV3CT.ext_bus.mapper	PVBusMapper
ARMNeoverseV3CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder
ARMNeoverseV3CT.global_debug_rom	debug_rom
ARMNeoverseV3CT.secondary_debug_rom	debug_rom
ARMNeoverseV3CT.sve	ScalableVectorExtension

This model has the following MTI trace components:

Name	Component type
ARMNeoverseV3CT.AMU	PVBusLogger
ARMNeoverseV3CT.AMU.mapper	PVBusMapper
ARMNeoverseV3CT.DAP	PVBusLogger
ARMNeoverseV3CT.DAP.mapper	PVBusMapper
ARMNeoverseV3CT.DSU	DSU-120
ARMNeoverseV3CT.DSU.PPU_cluster	PPUv1
ARMNeoverseV3CT.DSU.PPU_cluster.busslave	PVBusSlave
ARMNeoverseV3CT.DSU.PPU_core0	PPUv1
ARMNeoverseV3CT.DSU.PPU_core0.busslave	PVBusSlave
ARMNeoverseV3CT.DSU.mpam_busslave	PVBusSlave



Name	Component type
ARMNeoverseV3CT.DSU.shared_cache	PVCache
ARMNeoverseV3CT.DSU.shared_cache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseV3CT.DSU.shared_cache.upstream[Y] (where Y = 0–4)	PVBusSlave
ARMNeoverseV3CT.DSU.utility_slave[0]	PVBusSlave
ARMNeoverseV3CT.MMAP	PVBusLogger
ARMNeoverseV3CT.MMAP.mapper	PVBusMapper
ARMNeoverseV3CT.RAS	PVBusLogger
ARMNeoverseV3CT.RAS.mapper	PVBusMapper
ARMNeoverseV3CT.cpu0	ARM_Neoverse-V3
ARMNeoverseV3CT.cpu0.UTLB	TLB
ARMNeoverseV3CT.cpu0.l1dcache	PVCache
ARMNeoverseV3CT.cpu0.l1dcache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseV3CT.cpu0.l1dcache.upstream[0]	PVBusSlave
ARMNeoverseV3CT.cpu0.l1licache	PVCache
ARMNeoverseV3CT.cpu0.l1licache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseV3CT.cpu0.l1licache.upstream[0]	PVBusSlave
ARMNeoverseV3CT.cpu0.l2cache	PVCache
ARMNeoverseV3CT.cpu0.l2cache.downstream[0].pvbusmaster	PVBusMaster
ARMNeoverseV3CT.cpu0.l2cache.upstream[U] (where U = 0–1)	PVBusSlave
ARMNeoverseV3CT.ext_bus	PVBusLogger
ARMNeoverseV3CT.ext_bus.mapper	PVBusMapper
ARMNeoverseV3CT.gic_cpuif_decoder_cluster	GICv3CPUInterfaceDecoder

### Ports for ARMNeoverseV3CT

Port	Direction	Protocol	Description
acp_s	slave	PVBus	AXI ACP slave port.
AEND0MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND1MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND2MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
AEND3MP	slave	Value_64	Port to obtain end address of valid peripheral address range (exclusive).
ASTART0MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART1MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART2MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
ASTART3MP	slave	Value_64	Port to obtain start address of valid peripheral address range (inclusive).
broadcastatomic	slave	Signal	CHI defined pins.

Port	Direction	Protocol	Description
broadcastcachemaint	slave	Signal	ACE defined pins.
broadcastrouter	slave	Signal	ACE defined pins.
broadcastpersist	slave	Signal	CHI defined pins.
cfgend	slave	Signal	This signal is for EE bit initialisation.
cfgsdisable	slave	Signal	This signal disables write access to some secure Interrupt Controller registers.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the cluster level components run e.g cluster level timers, caches and pmu.
cluster_pcs_m_pchannel	master	PChannel	Cluster PCSM signal
cluster_powerdown_out	master	Signal	DEPRECATED - An output from PPU that informs thermal controller of the cluster power info
cluster_ppu_hw_stat	master	Value	Cluster PPU power state
clusterid	slave	Value	The cluster ID maps to the affinity levels of the MPIDR_EL1 register and is evaluated based on the MPIDR_EL1 layout. If MPIDR_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR_EL1 supports 24-bit cluster affinity levels, bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set ManagerID64 of the core.
clusterpmuirq	master	Signal	DynamiQ pmu irq
CNTHPIRQ	master	Signal	Timer signals to SOC.
CNTHPSIRQ	master	Signal	Timer signals to SOC
CNTHVIRQ	master	Signal	Timer signals to SOC.
CNTHVSIIRQ	master	Signal	Timer signals to SOC
CNTPNSIRQ	master	Signal	Timer signals to SOC.
CNTPSIIRQ	master	Signal	Timer signals to SOC.
cntvalueb	slave	CounterInterface	Interface to SoC level counter module.
CNTVIRQ	master	Signal	Timer signals to SOC.
commirq	master	Signal	Interrupt signal from debug communications channel.
core_clk_in	slave	ClockSignal	The clock signal connected to the core_clk_in port is used to determine the rate at which each core executes instructions.
core_pcs_m_pchannel	master	PChannel	Core PCSM signals
core_powerdown_out	master	Signal	DEPRECATED - An output from PPU that informs thermal controller of the core power info
core_ppu_hw_stat	master	Value	Notify the power state change inside the PPU.
coreerrirq	master	Signal	Error indicator for an ECC error that causes potential data corruption or loss of coherency
corefaultirq	master	Signal	Fault indicator for a detected 1 or 2 bit ECC error
coreinstrrun	master	Signal	Core Running State

Port	Direction	Protocol	Description
cryptodisable	slave	Signal	Disable cryptography extensions after reset.
cti	master	v8EmbeddedCrossTrigger_controlprotocol	Cross trigger matrix port.
ctidbgirq	master	Signal	Cross Trigger Interface (CTI) interrupt trigger output.
dbgen	slave	Signal	External debug interface.
dbgnopwrdown	master	Signal	No power-down request.
dbgpwrupreq	master	Signal	Debug power up request.
defaultMP	slave	Signal	If DefaultMP is asserted, all the transactions go to peripheral ports except the configured address ranges. Configured address ranges are sent to main master interface ports.
dev_debug_s	slave	PVBus	External debug interface.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE.
fiq	slave	Signal	This signal drives the CPUs fast-interrupt handling.
gicreset	master	Signal	An output from PPU that can be used to reset GICCLK domain components
gicv3_redistributor_s	slave	GICv3Comms	GICv3 AXI-stream port.
irq	slave	Signal	This signal drives the CPUs interrupt handling.
l0gptsz	slave	Value	RME LOGPTSZ port
legacy_tz_en	slave	Signal	RME LEGACY_TZ_EN port
memorymapped_debug_s	slave	PVBus	External debug interface.
pmbirq	master	Signal	Interrupt signal from the statistical profiling unit.
pmuirq	master	Signal	Interrupt signal from performance monitoring unit.
ppu_cluster_irq	master	Signal	PPU cluster interrupt.
ppu_cluster_isolate	master	Signal	PPU Cluster Domain isolation control.
ppu_cluster_wakerequest	slave	Signal	PPU cluster wake request signal.
ppu_core_irq	master	Signal	PPU core interrupt.
ppu_core_isolate	master	Signal	PPU Core Domain isolation control.
ppu_core_wakerequest	slave	Signal	PPU core wake request signal.
presetdbg	slave	Signal	Initialize the shared debug APB, Cross Trigger Interface (CTI), and Cross Trigger Matrix (CTM) logic.
pvbus_m0	master	PVBus	The core will generate bus requests on this port.
pvbus_periph_m	master	PVBus	The core can generate peripheral bus request on this port.
rei	slave	Signal	Per core RAM Error Interrupt.
reset	slave	Signal	Reset signal to cluster.
rlpiden	slave	Signal	External debug interface.
rtpiden	slave	Signal	External debug interface.
rvbaraddr	slave	Value_64	Reset vector base address.
sci_m	master	SystemCoherencyInterface	System coherency interface port, which is used to take the whole cluster into/out-of coherency domain

Port	Direction	Protocol	Description
sei	slave	Signal	Per core System Error physical pins.
spiden	slave	Signal	External debug interface.
ticks	master	InstructionCount	This port should be connected to one of the two ticks ports on a 'visualisation' component, in order to display a running instruction count.
trbirq	master	Signal	Interrupt signal from the trace buffer unit.
utility_bus_s	slave	PVBus	Utility bus slave
vcpumntirq	master	Signal	Interrupt signal for virtual CPU maintenance IRQ.
vfiq	slave	Signal	Virtualised FIQ.
virq	slave	Signal	Virtualised IRQ.
virtio_s	slave	PVBus	The virtio coherent port, hooks directly into the L2 system and becomes coherent (assuming attributes are set correctly).
vsei	slave	Signal	Per core virtual System Error physical pins.

## Parameters for ARMNeoverseV3CT

### **cpu0 . CFGEND**

Endianness configuration at reset. 0, little endian. 1, big endian. Big endian is unsupported in the presence of Future Architecture Technologies (FAT).

Type: `bool`

Default value: `false`

### **cpu0 . CFGTE**

Instruction set state when resetting into AArch32. 0, A32. 1, T32.

Type: `bool`

Default value: `false`

### **cpu0 . CRYPTODISABLE**

Disable cryptographic features.

Type: `bool`

Default value: `false`

### **cpu0 . RVBARADDR**

Value of RVBAR\_ELx register.

Type: `uint64_t`

Default value: `0x0`

**cpu0.crypto\_aes**

AES instructions supported (requires CryptoPlugin to be loaded). 0, not implemented. 2, AES and PMULL instructions implemented (FEAT\_AES, FEAT\_PMULL).

Type: `uint8_t`

Default value: 2

**cpu0.crypto\_sha3**

Implement ARMv8.4 SHA-3 instructions (requires CryptoPlugin to be loaded) (FEAT\_SHA3).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 2

**cpu0.crypto\_sha512**

Implement ARMv8.4 SHA-512 instructions (requires CryptoPlugin to be loaded) (FEAT\_SHA512).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 2

**cpu0.crypto\_sm3**

Implement ARMv8.4 SM-3 instructions (requires CryptoPlugin to be loaded) (FEAT\_SM3).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 2

**cpu0.crypto\_sm4**

Implement ARMv8.4 SM-4 instructions (requires CryptoPlugin to be loaded) (FEAT\_SM4).values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 2

**cpu0.enable\_trace\_special\_hlt\_imm16**

Enable usage of parameter trace\_special\_hlt\_imm16.

Type: `bool`

Default value: `false`

### **`cpu0.l2cache-hit_latency`**

L2 Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`cpu0.l2cache-maintenance_latency`**

L2 Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`cpu0.l2cache-miss_latency`**

L2 Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`cpu0.l2cache-read_access_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`cpu0.l2cache-read_latency`**

L2 Cache timing annotation latency for read accesses given in ticks per byte accessed. `l2cache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`cpu0.l2cache-size`**

L2 Cache size in bytes.

Type: `uint32_t`

Default value: `0x200000`

### **`cpu0.l2cache-snoop_data_transfer_latency`**

L2 Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`cpu0.l2cache-snoop_issue_latency`**

L2 Cache timing annotation latency for snoop accesses issued by this cache in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`cpu0.l2cache-ways`**

L2 Cache number of ways (sets are implicit from size).

Type: `uint8_t`

Default value: `8`

### **`cpu0.l2cache-write_access_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per access. If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `l2cache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`cpu0.l2cache-write_latency`**

L2 Cache timing annotation latency for write accesses given in ticks per byte accessed. `l2cache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`cpu0.max_code_cache_mb`**

Maximum size of the simulation code cache (MiB). For platforms with more than 2 cores this limit will be scaled down. (e.g 1/8 for 16 or more cores).

Type: `uint16_t`

Default value: `0x100`

### **`cpu0.min_sync_level`**

Force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

### **`cpu0.semihosting-A32_HLT`**

A32 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`cpu0.semihosting-A64_HLT`**

A64 HLT number for semihosting calls.

Type: `uint16_t`

Default value: `0xf000`

### **`cpu0.semihosting-ARM_SVC`**

A32 SVC number for semihosting calls.

Type: `uint32_t`

Default value: `0x123456`

### **`cpu0.semihosting-T32_HLT`**

T32 HLT number for semihosting calls.

Type: `uint8_t`

Default value: `60`

### **`cpu0.semihosting-Thumb_SVC`**

T32 SVC number for semihosting calls.

Type: `uint8_t`

Default value: `171`

### **`cpu0.semihosting-cmd_line`**

Command line available to semihosting calls.



Type: `string`

Default value: `N/A`

### **`cpu0.semihosting-cwd`**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

### **`cpu0.semihosting-enable`**

Enable semihosting SVC/HLT traps.

Type: `bool`

Default value: `true`

### **`cpu0.semihosting-heap_base`**

Virtual address of heap base.

Type: `uint64_t`

Default value: `0x0`

### **`cpu0.semihosting-heap_limit`**

Virtual address of top of heap.

Type: `uint64_t`

Default value: `0xf000000`

### **`cpu0.semihosting-stack_base`**

Virtual address of base of descending stack.

Type: `uint64_t`

Default value: `0x10000000`

### **`cpu0.semihosting-stack_limit`**

Virtual address of stack limit.

Type: `uint64_t`

Default value: `0xf000000`

**cpu0.trace\_special\_hlt\_imm16**

For this HLT number, IF enable\_trace\_special\_hlt\_imm16=true, skip performing usual HLT execution but call MTI trace if registered.

Type: uint16\_t

Default value: 0xf000

**cpu0.vfp-enable\_at\_reset**

Enable VFP in CPACR, CPPWR, NSACR at reset. Warning: Arm recommends going through the implementation's suggested VFP power-up sequence!.

Type: bool

Default value: false

**cpu0.vfp-present**

Set whether the model has VFP support; always true in the presence of Future Architecture Technologies (FAT).

Type: bool

Default value: true

**AEND0\_DEFAULT**

Default end address for peripheral port 0 address range exclusive (corresponds to AENDMP0 input signal).

Type: uint64\_t

Default value: 0x0

**AEND1\_DEFAULT**

Default end address for peripheral port 1 address range exclusive (corresponds to AENDMP1 input signal).

Type: uint64\_t

Default value: 0x0

**AEND2\_DEFAULT**

Default end address for peripheral port 2 address range exclusive (corresponds to AENDMP2 input signal).

Type: uint64\_t

Default value: 0x0

**AEND3\_DEFAULT**

Default end address for peripheral port 3 address range exclusive (corresponds to AENDMP3 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART0\_DEFAULT**

Default start address for peripheral port 0 address range inclusive (corresponds to ASTARTMP0 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART1\_DEFAULT**

Default start address for peripheral port 1 address range inclusive (corresponds to ASTARTMP1 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART2\_DEFAULT**

Default start address for peripheral port 2 address range inclusive (corresponds to ASTARTMP2 input signal).

Type: `uint64_t`

Default value: `0x0`

**ASTART3\_DEFAULT**

Default start address for peripheral port 3 address range inclusive (corresponds to ASTARTMP3 input signal).

Type: `uint64_t`

Default value: `0x0`

**BROADCASTATOMIC**

Enable broadcasting of atomic operation. The broadcastatomic signal will override this value if used.

Type: `bool`

Default value: `true`

**BROADCASTCACHEMAINT**

Enable broadcasting of cache maintenance operations to downstream caches. The broadcastcachemaint signal will override this value if used.

Type: `bool`

Default value: `false`

**BROADCASTOUTER**

Enable broadcasting of Outer Shareable transactions. The broadcastouter signal will override this value if used.

Type: `bool`

Default value: `false`

**BROADCASTPERSIST**

Enable broadcasting of cache clean to the point of persistence operations. The broadcastpersist signal will override this value if used.

Type: `bool`

Default value: `true`

**CLUSTER\_ID**

The Cluster ID maps to the affinity levels of the MPIDR\_EL1 register and is evaluated based on the MPIDR\_EL1 layout. If MPIDR\_EL1 supports 16-bit cluster affinity levels, bits [15:8] map to IDRAFF3, while bits [7:0] map to IDRAFF2. If MPIDR\_EL1 supports 24-bit cluster affinity levels, the bits [23:16] map to IDRAFF3, bits [15:8] map to IDRAFF2, and bits [7:0] map to IDRAFF1. This configuration also updates all relevant component DEVAFF registers and is used to set the ManagerID64 of the core.

Type: `uint32_t`

Default value: `0x0`

**CMO\_broadcast\_when\_cache\_state\_modelling\_disabled**

Skip broadcasting some cache maintenance operations (CMOs) when cache state modelling is disabled as a performance optimisation. this value may affect performance of the simulation, but is required if the rest of the system is sensitive to certain even when cache state modelling is disabled. Possible values are:- 0 = All CMOs are broadcast if architecturally required- 1 = Data cache maintenance to the PoC or PoU are not broadcast when data cache state modelling is disabled. Otherwise all CMOs are broadcast if architecturally required.

Type: `uint8_t`

Default value: `1`

**CPUCFR**

Value of CPU Configuration Register.

Type: `uint64_t`

Default value: `0x20`

**GICDISABLE**

Disable the new style GICv3 CPU interface in each core model. It can be set for the platform not having a GICv3 Interface (GICv3).

Type: `bool`

Default value: `true`

**NUM\_CORES**

Number of cores per cluster.

Type: `uint8_t`

Default value: `1`

**brbe\_disable\_recording**

If BRBE is implemented and this is set to true, disable BRBE recording. All registers will be functional, but no branches will be recorded. This will improve model performance for workloads that enable BRBE, but don't care about the information stored in it. (FEAT\_BRBE).

Type: `bool`

Default value: `true`

**brbe\_log2\_num\_records**

Log2 of number of BRB records supported. 3 -> 8 records, ... 6 -> 64 records.

Type: `uint8_t`

Default value: `5`

**bus\_type**

Cosmetic change that changes reset value of L2ACTLR register. 0, ACE. 1, CHI. 2, AXI.

Type: `uint8_t`

Default value: `0`

**core\_power\_on\_by\_default**

If true, The cluster and cores will be powered on for v9 after reset. If this parameter is enabled, the powerdown signal from in-cluster PPUs which is connected to cores and cluster reset signals will be cleared after reset and triggers the power on sequence in PPU.

Type: `bool`

Default value: `false`

**cpi\_div**

Divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**cpi\_mul**

Multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**dcache-hit\_latency**

L1 D-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-maintenance\_latency**

L1 D-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-miss\_latency**

L1 D-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**dcache-prefetch\_enabled**

Enable simulation of data cache prefetching. This is only used when dcache-state\_modelled=true.

Type: `bool`

Default value: `false`

**dcache-read\_access\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per access (of size dcache-read\_bus\_width\_in\_bytes). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if dcache-read\_latency is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**dcache-read\_latency**

L1 D-Cache timing annotation latency for read accesses given in ticks per byte accessed. dcache-read\_access\_latency must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**dcache-size**

L1 D-Cache size in bytes.

Type: `uint32_t`

Default value: `0x10000`

**dcache-snoop\_data\_transfer\_latency**

L1 D-Cache timing annotation latency for received snoop accesses that perform a data transfer given in ticks per byte accessed. This is only used when dcache-state\_modelled=true.

Type: `uint64_t`

Default value: `0x0`

**dcache-state\_modelled**

Set whether D-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **`dcache-write_access_latency`**

L1 D-Cache timing annotation latency for write accesses given in ticks per access (of size `dcache-write_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `dcache-write_latency` is set. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`dcache-write_latency`**

L1 D-Cache timing annotation latency for write accesses given in ticks per byte accessed. `dcache-write_access_latency` must be set to 0 for per-byte latencies to be applied. This is only used when `dcache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

### **`default_opmode`**

Operating mode of DynamIQ coming out of reset. 0: SFONLY ON, 1: 1/4 CACHE ON, 2: 1/2 CACHE ON, 3: 3/4 CACHE ON, 4: FULL CACHE ON.

Type: `uint8_t`

Default value: 4

### **`diagnostics`**

Enable DynamIQ diagnostic messages.

Type: `bool`

Default value: `false`

### **`ecv_support_level`**

Implement Enhanced Counter Virtualization feature from ARMv8.6. 0, Not supported. 1, fully supported without CNTPOFF. 2, fully supported with CNTPOFF (FEAT\_ECV).

Type: `uint8_t`

Default value: 2

### **`enable_simulation_performance_optimizations`**

With this option enabled, the model will run more quickly, but be less accurate to exact CPU behavior. The model will still be functionally accurate for software, but may increase differences



seen between hardware behavior and model behavior for certain workloads (it changes the micro-architectural value of `stage12_tlb_size` parameter to 1024).

Type: `bool`

Default value: `true`

### **ete.CLAIMTAGS**

Number of claim tags.

Type: `uint8_t`

Default value: 4

### **ete.MAX\_INST\_PER\_Q**

Maximum limit for the number of instructions implied by a Q element.

Type: `uint16_t`

Default value: `0x1`

### **ete.NumberOfRSPairs**

Number of resource selector pairs.

Type: `uint8_t`

Default value: 8

### **ete.PIDR\_CMOD**

TRCPIDR CMOD value.

Type: `uint8_t`

Default value: 0

### **ete.Q\_CADENCE**

Number of instruction blocks traced between two Q elements.

Type: `uint16_t`

Default value: `0x1`

### **ete.RES0\_STATEFUL**

Whether **RES0** bits are stateful or **RAZ/WI**.

Type: `bool`

Default value: `false`

**ete.RETSTACK**

Return stack depth.

Type: `uint8_t`

Default value: 3

**ete.SIM\_OVERFLOW\_GRANULARITY**

Number of instruction blocks in each granule, for simulated overflow.

Type: `uint32_t`

Default value: `0x64`

**ete.SIM\_OVERFLOW\_PERCENTAGE**

Percentage of instruction blocks lost in each granule, for simulated overflow.

Type: `uint8_t`

Default value: 0

**ete.SOURCE\_ADDRESS**

Allow generation of source address elements.

Type: `bool`

Default value: `false`

**ete.TRACE\_OUTPUT**

File to which to write trace byte stream.

Type: `string`

Default value: N/A

**ete.TRCSRSTA\_FORCED\_EXCEP**

TRCSRSTA value for a forcibly traced exception.

Type: `bool`

Default value: `false`

**ext\_abort\_so\_write\_ras\_type**

External Aborts are reported as RAS error type specified in this param. Values: 0 = NONE, 1 = UC, 2 = UEU, 3 = UEO, 4 = UER, 5 = CE.

Type: `uint8_t`

Default value: 2

### **force\_mte\_tag\_access\_razwi\_and\_ignore\_tag\_checks**

Force MTE tag accesses to **RAZ/WI** and also ignore tag checks irrespective of whether memory is tagged or not. This parameter doesn't affect non-load/store tag generation instructions like ADDG/GMI/IRG/SUBG. Please note that setting this parameter to true will also disable MTE tag access and tag check related traces.

Type: `bool`

Default value: `false`

### **force\_zero\_PSTATE\_PAN**

Non-architecture parameter to force PSTATE.PAN to be 0.0: No effect. 1: PSTATE.PAN is always treated as 0. The parameter optimizes the performance of updating PSTATE.PAN.

Type: `bool`

Default value: `false`

### **force\_zero\_mpam\_partid\_and\_pmg**

Non-architecture parameter to force MPAM PARTID, PMG and NS/SP to 0.0: No effect. 1: PARTID and PMG are always treated as 0. The parameter optimizes the performance of updating the registers MPAMO\_EL1, MPAM1\_EL1, MPAM2\_EL2, MPAM3\_EL3, MPAMHCR\_EL2.

Type: `uint8_t`

Default value: 0

### **has\_coherent\_icache**

Whether icache invalidation to the point of unification is required for instruction to data coherence.  
true - Invalidate operations not required.

Type: `bool`

Default value: `true`

### **has\_delayed\_dbgreg**

Delay the functional effect of external debug register writes until ISB or implicit barrier.

Type: `bool`

Default value: `false`

### **has\_delayed\_sysreg**

Delay the functional effect of system register writes until ISB or implicit barrier.

Type: `bool`

Default value: `true`

### **has\_enhanced\_pan**

Implements Armv8.7 Enhanced PAN feature (FEAT\_PAN3) values of this parameter are:- 1, feature is implemented if ARMv8.7 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 2

### **has\_ete**

If true, implements the Embedded Trace Extension (FEAT\_ETE). This option is discarded if ete plugin is explicitly loaded (-plugin or -P).

Type: `bool`

Default value: `false`

### **has\_peripheral\_port**

If true, an additional AXI peripheral port is configured.

Type: `bool`

Default value: `false`

### **has\_rndr**

Implement random number instructions to read from RNDR and RNDRSS random number registers from ARMv8.5 (FEAT\_RNG). values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.5 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

### **has\_statistical\_profiling**

Whether Statistical Based Profiling is implemented (FEAT\_SPE).

Type: `bool`

Default value: `true`

### **has\_v8\_7\_spe\_inverted\_filtering**

Where FEAT\_SPEv1p2 is implemented, whether inverted filtering by events is implemented (represented by PMISDR.FnE).

Type: `bool`

Default value: `false`

**has\_v8\_7\_spe\_previous\_branch\_target**

Where FEAT\_SPEv1p2 is implemented, whether the optional branch target feature is implemented (FEAT\_SPE\_PBT).

Type: `bool`

Default value: `false`

**icache-hit\_latency**

L1 I-Cache timing annotation latency for hit. Intended to model the tag-lookup time. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-maintenance\_latency**

L1 I-Cache timing annotation latency for cache maintenance operations given in total ticks. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-miss\_latency**

L1 I-Cache timing annotation latency for miss. Intended to model the time for failed tag-lookup and allocation of intermediate buffers. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: `0x0`

**icache-prefetch\_enabled**

Enable simulation of instruction cache prefetching. This is only used when `icache-state_modelled=true`.

Type: `bool`

Default value: `false`

**icache-read\_access\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per access (of size `icache-read_bus_width_in_bytes`). If this parameter is non-zero, per-access latencies will be used instead of per-byte even if `icache-read_latency` is set. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus, this is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **icache-read\_latency**

L1 I-Cache timing annotation latency for read accesses given in ticks per byte accessed. `icache-read_access_latency` must be set to 0 for per-byte latencies to be applied. This is in addition to the hit or miss latency, and intended to correspond to the time taken to transfer across the cache upstream bus. This is only used when `icache-state_modelled=true`.

Type: `uint64_t`

Default value: 0x0

### **icache-size**

L1 I-Cache size in bytes.

Type: `uint32_t`

Default value: 0x10000

### **icache-state\_modelled**

Set whether I-cache has stateful implementation.

Type: `bool`

Default value: `false`

### **instruction\_tlb\_size**

Number of stage1+2 itlb entries (or 0 for unified ITLB+DTLB).

Type: `uint32_t`

Default value: 0x0

### **invalidate\_code\_cache\_on\_icache\_cmo**

If set, all PEs will invalidate simulation code cache when an I-side CMO is executed in a PE. Note: Enabling this parameter will reduce simulation performance.

Type: `uint8_t`

Default value: 0

### **l3cache-ways**

L3 Cache number of ways (sets are implicit from size).

Type: `uint16_t`

Default value: 16

**log2\_trace\_buffer\_alignment**

Log2 of trace buffer alignment constraint for output buffer (0->1B ... 11->2Kib).

Type: `uint8_t`

Default value: 6

**memory\_tagging\_support\_level**

Specify the memory tagging extension support level: 0, not implemented.1, instructions and registers only are implemented (FEAT\_MTE).2, implemented (FEAT\_MTE2).3, implemented with asymmetric handling of exceptions (FEAT\_MTE3).

Type: `uint8_t`

Default value: 3

**mpam\_has\_altsp**

DEPRECATED: MPAMIDR\_EL1.HAS\_ALTSP is required when FEAT\_RME is implemented.

Type: `bool`

Default value: `false`

**mpamidr\_has\_force\_ns**

Whether MPAMIDR\_EL1.HAS\_FORCE\_NS bit is set or clear.values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.6 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

**mpamidr\_has\_sdeflt**

Whether MPAMIDR\_EL1.HAS\_SDEFLT bit is set or clear.values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.6 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

**mpamidr\_has\_tidr**

Whether MPAMIDR\_EL1.HAS\_TIDR bit is set or clear.values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.6 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 1

**mpmm\_accumulator\_multiplier**

Parameter for non-architectural behaviour to provide a multiplier for the MPMM accumulator.value of n means the accumulator will use (n \* accumulator value) to calculate the mpmm threshold (MPMM).is provided as a fast model workaround to handle cases where execution of quantum in SystemC does not result in an expected threshold in MPMM counters being reached at the anticipated time.

Type: uint8\_t

Default value: 1

**ptw\_latency**

Page table walker latency for TA (Timing Annotation), expressed in simulation ticks.

Type: uint64\_t

Default value: 0x0

**reported\_patch\_level**

Purely cosmetic patch level value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: int8\_t

Default value: -1

**reported\_revision\_number**

Purely cosmetic revision number value to be displayed from the ID registers. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: int8\_t

Default value: -1

**rme\_level0\_gpt\_size**

The range of address space protected by each entry in the level 0 GPT (0->1GB 1->16GB, 2->64GB, 3->512GB).

Type: uint8\_t

Default value: 0

**rme\_support\_level**

0 -> Realm management extension not implemented, 1 -> LEGACY\_TZ\_EN mode i.e. RME register fields are stateful but only supports secure/non-secure states, 2 -> Realm management extension fully implemented (FEAT\_RME).

Type: uint8\_t



Default value: 2

**rndr\_rndrrs\_seed**

Initial seed for random engine used in RNDR register.

Type: `uint64_t`

Default value: `0x0`

**stage12\_tlb\_size**

Number of stage1+2 tlb entries. Valid values are 0 or  $\geq 4$ .

Type: `uint32_t`

Default value: `0x80`

**tcr\_txsz\_undersize\_should\_fault**

If large VA is not supported, Whether undersized TxSZ value should generate translation fault.

Type: `bool`

Default value: `false`

**tlb\_latency**

TLB latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

**tlbi\_stall\_enabled**

If true, tlb invalidation broadcast requests will block the requesting PE until all the other PEs flush their TLBs.

Type: `bool`

Default value: `false`

**treat\_PAC\_as\_NOP**

Non-architecture parameter. Treat Pointer Authentication as **NOP**. When the parameter is true behaviour of FEAT\_PAuth instructions changes as following PAC\* and AUT\* and XPAC\* instructions are **NOP**. Brach, load, return with authentication instructions do not check and change signature bits and ignore FEAT\_PAuth traps.

Type: `bool`

Default value: `false`

**walk\_cache\_latency**

Walk cache latency for TA (Timing Annotation), expressed in simulation ticks.

Type: `uint64_t`

Default value: `0x0`

## 3.112 ARMSC000CT

Defined in `LISA/ARMSC000CT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p1	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following ports were removed:

- `currpri`

### Differences between the model and the RTL

The model has the following limitations:

- It does not implement any security features.
- Only bit[0] of the Auxiliary Control Register is supported for read/write. No functionality is implemented.
- The Security Features Control Register read/write access is supported using `SECKEY`. No functionality is implemented.

### Iris and MTI instances for ARMSC000CT

This model has the following Iris instances:

Name	Instance type
ARMSC000CT	ARM_SC000
ARMSC000CT.acp_mapper	PVBusMapper
ARMSC000CT.ext_bus	PVBusLogger
ARMSC000CT.ext_bus.mapper	PVBusMapper
ARMSC000CT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

Name	Component type
ARMSC000CT	ARM_SC000
ARMSC000CT.acp_mapper	PVBusMapper
ARMSC000CT.ext_bus	PVBusLogger
ARMSC000CT.ext_bus.mapper	PVBusMapper
ARMSC000CT.l2_flusher	AsyncCacheFlushUnit

## Ports for ARMSC000CT

Port	Direction	Protocol	Description
ahbd	slave	PVBus	Debug AHB - core bus slave driven by the DAP.
bigend	slave	Signal	Configure big endian data format.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
edbgrq	slave	Signal	External debug request.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE and corresponds to the RTL TXEV and RXEV signals.
intisr	slave	Signal	This signal array delivers signals to the NVIC.
intnmi	slave	Signal	Configure non maskable interrupt.
io_port_in	slave	PVBus	I/O port pair. See the documentation for the io_port_out port.
io_port_out	master	PVBus	I/O port pair. Used if IOP is true. Transactions from io_port_out which do not “match” should be returned via io_port_in. For performance reasons, the I/O port interface is not modelled directly. Instead, a simple PVBus gasket is inserted at the point in the memory system where the I/O port would be. In hardware, a device would be attached to the port which would tell the CPU whether it would like to intercept each transaction, given its address. This can be modelled in a performant manner by connecting a PVBusMapper-based device to io_port_out which intercepts transactions of interest and passes other transactions back to the CPU via io_port_in. Your I/O port device model is also responsible for aborting transactions which would be aborted on hardware (e.g. exclusives) if necessary.
lockup	master	Signal	Asserted when the processor is in lockup state.
poreset	slave	Signal	Raising this signal will do a power-on reset of the core.
pv_ppbus_m	master	PVBus	The core will generate External Private Peripheral Bus requests on this port.
pvbus_m	master	PVBus	The core will generate bus requests on this port.
sleepdeep	master	Signal	Asserted when the processor is in deep sleep.
sleeping	master	Signal	Asserted when the processor is in sleep.
stcalib	slave	Value	This is the calibration value for the SysTick timer The following bit flag is modelled stcalib[25] - NOREF - Indicates no reference clock integrated. If stclk is not in use, set to 1
stclk	slave	ClockSignal	This is the reference clock for the SysTick timer, called STCLKEN in the specification
sysreset	slave	Signal	Raising this signal will put the core into reset mode (but does not reset the debug logic).
sysresetreq	master	Signal	Asserted to indicate that a reset is required.
ticks	master	InstructionCount	Port allowing the number of instructions since startup to be read from the CPU.

## Parameters for ARMSC000CT

### BIGENDINIT

Initialize processor to big endian mode.

Type: `bool`

Default value: `false`

### **BKPT**

Number of breakpoint unit comparators implemented.

Type: `uint8_t`

Default value: 4

### **DBG**

Set whether debug extensions are implemented.

Type: `bool`

Default value: `true`

### **IOP**

Send all d-side transactions to the port, `io_port_out`. Transactions which do not match should be returned to the port, `io_port_in`.

Type: `bool`

Default value: `false`

### **IRQDIS**

IRQ line disable mask. Bit `n` of this 32-bit parameter disables `IRQ[n]`.

Type: `uint32_t`

Default value: `0x0`

### **NUM\_IRQ**

Number of user interrupts.

Type: `uint8_t`

Default value: 32

### **NUM\_MPU\_REGION**

Number of MPU regions.

Type: `uint8_t`

Default value: 0

**SYST**

Enable support for SysTick timer functionality.

Type: `bool`

Default value: `true`

**USER**

Enable support for Unprivileged/Privileged Extension.

Type: `bool`

Default value: `true`

**VTOR**

Include Vector Table Offset Register.

Type: `bool`

Default value: `true`

**WIC**

Include support for WIC-mode deep sleep.

Type: `bool`

Default value: `true`

**WPT**

Number of watchpoint unit comparators implemented.

Type: `uint8_t`

Default value: `2`

**`cpi_div`**

divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**`cpi_mul`**

multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**manager\_id**

Manager ID presented in bus transactions.

Type: `uint64_t`

Default value: `0x0`

**min\_sync\_level**

force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

**reported\_patch\_level**

Purely cosmetic patch level value to be displayed from the CUID register. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: `int8_t`

Default value: `-1`

**reported\_revision\_number**

Purely cosmetic revision number value to be displayed from the CUID register. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: `int8_t`

Default value: `-1`

**semihosting-Thumb\_SVC**

T32 SVC number for semihosting.

Type: `uint8_t`

Default value: `171`

**semihosting-cmd\_line**

Command line available to semihosting SVC calls.

Type: `string`

Default value: `N/A`

**semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: `N/A`

### **semihosting-enable**

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

Type: `bool`

Default value: `true`

### **semihosting-heap\_base**

Virtual address of heap base.

Type: `uint32_t`

Default value: `0x0`

### **semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint32_t`

Default value: `0x20700000`

### **semihosting-stack\_base**

Virtual address of base of descending stack.

Type: `uint32_t`

Default value: `0x20800000`

### **semihosting-stack\_limit**

Virtual address of stack limit.

Type: `uint32_t`

Default value: `0x20700000`

## **3.113 ARMSC300CT**

Defined in `LISA/ARMSC300CT.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p1	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

## Differences between the model and the RTL

The model has the following limitations:

- It does not implement any security features.
- The Trash Register is implemented as **RAZ/WI**.
- Only bit[0] of the Auxiliary Control Register is supported for read/write. No functionality is implemented.
- The Security Features Control Register read/write access is supported using **SECKEY**. No functionality is implemented.

## Implementation of ITM in M-class models

This model has a parameter that enables partial support for Instrumentation Trace Macrocell (ITM). In hardware or RTL, trace data from ITM is sent in packets to the trace block serially using a single pin or wire. In the model, if it is enabled, the ITM trace data is output using an MTI trace source called ITM. The ITM trace source has an **ITM\_PACKET\_TYPE** field. The following table shows which packet types the model supports:

Field value	Description	Supported by model
ITM_SYNC	Synchronization packet	Not supported.
ITM_P_OVERFLOW	Protocol: Overflow packet	Not supported.
ITM_P_LOCAL_TIMESTAMP	Protocol: Local timestamp packets	Not supported.
ITM_P_GLOBAL_TIMESTAMP	Protocol: Global timestamp packets	Not supported.
ITM_P_EXTEN	Protocol: Extension packet	Not supported.
ITM_S_INSTRUMENTATION	Source: Instrumentation packet	Supported.
ITM_S_DWT_EVENT_COUNTER	Hardware source: Event counter wrapping	Not supported.
ITM_S_DWT_EXCEPTION	Hardware source: Exception tracing	Supported.
ITM_S_DWT_PC_SAMPLING	Hardware source: PC sampling	Not supported.
ITM_S_DWT_DATA_PC_TRACE	Hardware source: DWT Data trace PC value	Supported.
ITM_S_DWT_DATA_ADDRESS_TRACE	Hardware source: DWT Data trace address value	Supported.
ITM_S_DWT_DATA_DATA_TRACE	Hardware source: DWT Data trace DATA value	Supported.

## Iris and MTI instances for ARMSC300CT

This model has the following Iris instances:

Name	Instance type
ARMSC300CT	ARM_SC300
ARMSC300CT.acp_mapper	PVBusMapper
ARMSC300CT.ext_bus	PVBusLogger
ARMSC300CT.ext_bus.mapper	PVBusMapper



Name	Instance type
ARMSC300CT.l2_flusher	AsyncCacheFlushUnit

This model has the following MTI trace components:

Name	Component type
ARMSC300CT	ARM_SC300
ARMSC300CT.acp_mapper	PVBusMapper
ARMSC300CT.ext_bus	PVBusLogger
ARMSC300CT.ext_bus.mapper	PVBusMapper
ARMSC300CT.l2_flusher	AsyncCacheFlushUnit

### Ports for ARMSC300CT

Port	Direction	Protocol	Description
ahbd	slave	PVBus	Debug AHB - core bus slave driven by the DAP.
auxfault	slave	Value	This is wired to the Auxiliary Fault Status Register.
bigend	slave	Signal	Configure big endian data format.
clk_in	slave	ClockSignal	The clock signal connected to the clk_in port is used to determine the rate at which the core executes instructions.
currpri	master	Value	Current execution priority.
edbgrq	slave	Signal	External debug request.
event	peer	Signal	This peer port of event input (and output) is for wakeup from WFE and corresponds to the RTL TXEV and RXEV signals.
intisr	slave	Signal	This signal array delivers signals to the NVIC.
intnmi	slave	Signal	Configure non maskable interrupt.
lockup	master	Signal	Asserted when the processor is in lockup state.
poreset	slave	Signal	Raising this signal will do a power-on reset of the core.
pv_ppbus_m	master	PVBus	The core will generate External Private Peripheral Bus requests on this port.
pdbus_m	master	PVBus	The core will generate bus requests on this port.
sleepdeep	master	Signal	Asserted when the processor is in deep sleep.
sleeping	master	Signal	Asserted when the processor is in sleep.
stcalib	slave	Value	This is the calibration value for the SysTick timer The following bit flag is modelled stcalib[25] - NOREF - Indicates no reference clock integrated. If stclk is not in use, set to 1
stclk	slave	ClockSignal	This is the reference clock for the SysTick timer.
sysreset	slave	Signal	Raising this signal will put the core into reset mode (but does not reset the debug logic).
sysresetreq	master	Signal	Asserted to indicate that a reset is required.
ticks	master	InstructionCount	Port allowing the number of instructions since startup to be read from the CPU.

### Parameters for ARMSC300CT

#### BB\_PRESENT

Enable bitbanding.

Type: bool

Default value: `true`

### **BIGENDINIT**

Initialize processor to big endian mode.

Type: `bool`

Default value: `false`

### **DBGLVL**

0: No debug; 1: Minimal debug; 2: Full debug without DWT address matching; 3: Full debug support with DWT data-comparators.

Type: `uint8_t`

Default value: 3

### **LVL\_WIDTH**

Number of bits of interrupt priority.

Type: `uint8_t`

Default value: 3

### **NUM\_IRQ**

Number of user interrupts.

Type: `uint8_t`

Default value: 16

### **NUM\_MPU\_REGION**

Number of MPU regions.

Type: `uint8_t`

Default value: 8

### **TRACE\_LVL**

Level of instrumentation trace supported. 0: No trace, 1: Standard trace. ITM, TPIU and DWT triggers and counters present, 2: Full trace. ITM, TPIU, ETM and DWT triggers and counters present.

Type: `uint8_t`

Default value: 1

**WIC**

Include support for WIC-mode deep sleep.

Type: `bool`

Default value: `true`

**cpi\_div**

divider for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**cpi\_mul**

multiplier for calculating CPI (Cycles Per Instruction).

Type: `uint32_t`

Default value: `0x1`

**manager\_id**

Manager ID presented in bus transactions.

Type: `uint64_t`

Default value: `0x0`

**min\_sync\_level**

force minimum syncLevel (0=off=default, 1=syncState, 2=postInsnIO, 3=postInsnAll).

Type: `uint8_t`

Default value: `0`

**reported\_patch\_level**

Purely cosmetic patch level value to be displayed from the CPUID register. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: `int8_t`

Default value: `-1`

**reported\_revision\_number**

Purely cosmetic revision number value to be displayed from the CPUID register. This value will not necessarily align with the model behaviour. The default value (-1) will see this parameter ignored.

Type: `int8_t`

Default value: -1

### **semihosting-Thumb\_SVC**

T32 SVC number for semihosting.

Type: `uint8_t`

Default value: 171

### **semihosting-cmd\_line**

Command line available to semihosting SVC calls.

Type: `string`

Default value: N/A

### **semihosting-cwd**

Base directory for semihosting file access.

Type: `string`

Default value: N/A

### **semihosting-enable**

Enable semihosting SVC traps. Applications that do not use semihosting must set this parameter to false.

Type: `bool`

Default value: `true`

### **semihosting-heap\_base**

Virtual address of heap base.

Type: `uint32_t`

Default value: `0x0`

### **semihosting-heap\_limit**

Virtual address of top of heap.

Type: `uint32_t`

Default value: `0x20700000`

**semihosting-stack\_base**  
Virtual address of base of descending stack.

Type: uint32\_t

Default value: 0x20800000

**semihosting-stack\_limit**  
Virtual address of stack limit.

Type: uint32\_t

Default value: 0x20700000

3.114 AddressTranslationUnit

Defined in LISA/AddressTranslationUnit.lisa.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
0.5	Alpha support

For an explanation of the quality levels, see [Quality level definitions](#).

**Changes in 11.31.15**  
The following parameters were added:

- ATUv2

**About AddressTranslationUnit**  
Address Translator Unit.

**Iris and MTI instances for AddressTranslationUnit**  
This model has the following Iris instances:

Name	Instance type
AddressTranslationUnit	AddressTranslationUnit
AddressTranslationUnit.ATU_BusMapper	PVBusMapper
AddressTranslationUnit.apb	PVBusSlave

This model has the following MTI trace components:

Name	Component type
AddressTranslationUnit.ATU_BusMapper	PVBusMapper

Name	Component type
AddressTranslationUnit.apb	PVBusSlave

### Ports for AddressTranslationUnit

Port	Direction	Protocol	Description
apb	slave	PVBus	-
irq_out	master	Signal	-
pvbus_m	master	PVBus	-
pvbus_s	slave	PVBus	-
reset_in	slave	Signal	-

### Parameters for AddressTranslationUnit

#### **ATUNTR**

Number of translation regions (1=2, 2=4, 3=8, 4=16, and 5=32).

Type: `uint8_t`

Default value: 5

#### **ATUPAW**

Physical address width (0=32, 1=36, 2=40, 3=44, 4=48, 5=52, 6=56, 7=64 bits). No impact on the PVBUS transactions.

Type: `uint8_t`

Default value: 5

#### **ATUPS**

Selects the page size granularity in bytes (0xC=4096, 0xD=8192, and 0xE=16384) (Default=0xD).

Type: `uint8_t`

Default value: 0xD

#### **ATUv2**

Have ATU follow the ATUv2 specification.

Type: `bool`

Default value: `false`

#### **DEBUG\_disable\_translation**

A model-only parameter to disable any remapping by the ATU, despite register configuration. For debug/test only.

Type: `bool`

Default value: `false`

### diagnostics

Show diagnostics messages.

Type: `uint32_t`

Default value: `0`

## 3.115 AndGate

Defined in `LISA/Gate.lisa`.

### About AndGate

This component implements a logical AND of two signal input ports to generate a single output signal. For example, you can use it to combine two interrupt signals.

### Iris and MTI instances for AndGate

This model has the following Iris instances:

Name	Instance type
AndGate	<a href="#">AndGate</a>

### Ports for AndGate

Port	Direction	Protocol	Description
input	slave	<a href="#">Signal</a>	2 input signals to be AND'ed.
output	master	<a href="#">Signal</a>	AND'ed output signal.

### Parameters for AndGate

This component does not have any parameters.

## 3.116 Ashbrook\_SoC\_SCC

Defined in `LISA/Ashbrook_SoC_SCC.lisa`.

### About Ashbrook\_SoC\_SCC

Ashbrook SoC Simple Configuration Controller IP Block.

### Iris and MTI instances for Ashbrook\_SoC\_SCC

This model has the following Iris instances:

Name	Instance type
Ashbrook_SoC_SCC	Ashbrook_SoC_SCC
Ashbrook_SoC_SCC.pvbusslave	PVBusSlave

This model has the following MTI trace components:

Name	Component type
Ashbrook_SoC_SCC.pvbusslave	PVBusSlave

### Ports for Ashbrook\_SoC\_SCC

Port	Direction	Protocol	Description
cluster_temperature	slave	ValueState	-
pvbus_s	slave	PVBus	-

### Parameters for Ashbrook\_SoC\_SCC

#### diagnostics

Diagnostics.

Type: uint32\_t

Default value: 0

#### gpr0

General Purpose Register 0.

Type: uint32\_t

Default value: 0

#### gpr1

General Purpose Register 1.

Type: uint32\_t

Default value: 0

#### gpr10

General Purpose Register 10.

Type: uint32\_t

Default value: 0

#### gpr11

General Purpose Register 11.



Type: `uint32_t`

Default value: 0

### **gpr12**

General Purpose Register 12.

Type: `uint32_t`

Default value: 0

### **gpr13**

General Purpose Register 13.

Type: `uint32_t`

Default value: 0

### **gpr14**

General Purpose Register 14.

Type: `uint32_t`

Default value: 0

### **gpr15**

General Purpose Register 15.

Type: `uint32_t`

Default value: 0

### **gpr2**

General Purpose Register 2.

Type: `uint32_t`

Default value: 0

### **gpr3**

General Purpose Register 3.

Type: `uint32_t`

Default value: 0

### **gpr4**

General Purpose Register 4.

Type: `uint32_t`

Default value: 0

### **`gpr5`**

General Purpose Register 5.

Type: `uint32_t`

Default value: 0

### **`gpr6`**

General Purpose Register 6.

Type: `uint32_t`

Default value: 0

### **`gpr7`**

General Purpose Register 7.

Type: `uint32_t`

Default value: 0

### **`gpr8`**

General Purpose Register 8.

Type: `uint32_t`

Default value: 0

### **`gpr9`**

General Purpose Register 9.

Type: `uint32_t`

Default value: 0

### **`num_clusters`**

Number of AP clusters in the platform.

Type: `uint32_t`

Default value: 12

### **`version`**

Version number of the platform.

Type: uint32\_t

Default value: 0

## 3.117 AsyncSignal

Defined in LISA/AsyncSignal.lisa.

### About AsyncSignal

This component provides the means to cleanly schedule events from non-simulation threads onto the simulation thread.

### Ports for AsyncSignal

Port	Direction	Protocol	Description
async_callback	master	AsyncSignalCallback	This port emits a call to signal() on the simulation thread asynchronously after async_control.signal() has been called.
async_control	slave	AsyncSignalControl	Non-simulation threads call signal() on this port in order to schedule an event: a call to async_callback.signal() on the simulation thread.

### Parameters for AsyncSignal

This component does not have any parameters.

## 3.118 AudioOut\_File

Defined in LISA/AudioOut\_File.lisa.

### About AudioOut\_File

This component implements an audio output that is suitable for use with the PL041\_AACI component. It writes raw 16-bit 48KHz stereo audio data to a user-specified file.

We expect this component to have little effect on the performance of PV systems. AudioOut\_File drains audio data at the rate that would be expected by software running in the simulation.

### Iris and MTI instances for AudioOut\_File

This model has the following Iris instances:

Name	Instance type
AudioOut_File	AudioOut_File

### Ports for AudioOut\_File

Port	Direction	Protocol	Description
audio	slave	AudioControl	Audio input for a connection to a component such as the PL041_AACI.

## Parameters for AudioOut\_File

### **fname**

Filename.

Type: `string`

Default value: `""`

## 3.119 AudioOut\_SDL

Defined in `LISA/AudioOut_SDL.lisa`.

### About AudioOut\_SDL

AudioOut\_SDL outputs audio using the host features of the Simple DirectMedia Layer (SDL) library.

This component results in SDL audio callbacks and might have a small impact on PV systems containing the component. It attempts to drain audio data at whatever rate is required to maintain smooth sound playback on the host PC. This might not match the data rate expected by applications running on the simulation.

### Iris and MTI instances for AudioOut\_SDL

This model has the following Iris instances:

Name	Instance type
AudioOut_SDL	AudioOut_SDL

### Ports for AudioOut\_SDL

Port	Direction	Protocol	Description
audio	slave	AudioControl	Audio input for a connection to a component such as the PL041_AACI.

### Parameters for AudioOut\_SDL

This component does not have any parameters.

## 3.120 BMU

Defined in `LISA/BMU.lisa`.

### About BMU

BMU (Bus Monitor Unit).

## Iris and MTI instances for BMU

This model has the following Iris instances:

Name	Instance type
BMU	BMU

## Ports for BMU

Port	Direction	Protocol	Description
apb_bus	slave	PVBus	for register read and write
bmu_input_bus	slave	PVBus	-
bus_trace_m	master	PVBus	-
clk_in	slave	ClockSignal	BMU core clock

## Parameters for BMU

### **amu\_element\_size**

AMU element size (0: 0KB, 1: 4KB).

Type: uint8\_t

Default value: 0

### **amu\_element\_start**

AMU element start.

Type: uint8\_t

Default value: 0

### **amu\_monitors**

Number of AMU monitors present.

Type: uint8\_t

Default value: 4

### **amu\_present**

Configures whether an AMU is included in the monitor.

Type: uint8\_t

Default value: 0

### **diagnostics**

Diagnostics.

Type: `uint8_t`

Default value: 2

### **feat\_rme**

RME support.

Type: `uint8_t`

Default value: 1

### **imu\_element\_size**

IMU element size (0: 0KB, 1: 4KB).

Type: `uint8_t`

Default value: 0

### **imu\_element\_start**

IMU element start.

Type: `uint8_t`

Default value: 0

### **monitor\_protocol**

Protocol 0:CHI, 1:AXI, 2:CXS.

Type: `uint8_t`

Default value: 0

### **mpam\_capture**

Determines if MPAM Capture interface is present.

Type: `uint8_t`

Default value: 0

### **mpam\_element\_size**

MPAM element size (0: 0KB, 1: 4KB, 2: 8KB, 3: 16KB).

Type: `uint8_t`

Default value: 0

### **mpam\_element\_start**

MPAM element start.

Type: `uint8_t`

Default value: 0

### **`mpam_monitors`**

Configures number of MPAM monitors present.

Type: `uint8_t`

Default value: 0

### **`mpam_present`**

Configures whether a MPAM is included in the monitor.

Type: `uint8_t`

Default value: 0

### **`mpam_scale_factor`**

Determines MPAM SCALE FACTOR (Powers of 2).

Type: `uint32_t`

Default value: 1

### **`mpam_width`**

Width of the MPAM properties.

Type: `uint8_t`

Default value: 12

### **`num_imu_monitors`**

Number of IMU monitors.

Type: `uint8_t`

Default value: 0

### **`num_monitors`**

Number of monitors in PMU.

Type: `uint8_t`

Default value: 1

### **`pm_dual_page_ext`**

Dual-page extension enabled.

Type: `uint8_t`

Default value: 0

### **`pm_edgedetect_ext`**

Edge detect extension enabled.

Type: `uint8_t`

Default value: 0

### **`pm_export_ext`**

Export extension enabled.

Type: `uint8_t`

Default value: 0

### **`pm_fzo_ext`**

Freeze on overflow extension enabled.

Type: `uint8_t`

Default value: 0

### **`pm_mpam_filter_ext`**

MPAM filtering extension enabled.

Type: `uint8_t`

Default value: 0

### **`pm_oac_ext`**

Observability and access control.

Type: `uint8_t`

Default value: 0

### **`pm_sos_filter_ext`**

Secure operating state filtering extension enabled.

Type: `uint8_t`

Default value: 1

### **`pm_sshot_ext`**

Snapshot extension enabled.



Type: `uint8_t`

Default value: 1

### **`pm_threshold_ext`**

Threshold extension enabled.

Type: `uint8_t`

Default value: 0

### **`pm_tro_ext`**

Trace generation extension enabled.

Type: `uint8_t`

Default value: 0

### **`pmevfiltr2_present`**

Event filtering register present.

Type: `uint8_t`

Default value: 0

### **`pmevfiltr_present`**

Event filtering register present.

Type: `uint8_t`

Default value: 0

### **`pmimpdef_present`**

Implementation defined register present.

Type: `uint8_t`

Default value: 0

### **`pmoverflow_present`**

Overflow interrupt present.

Type: `uint8_t`

Default value: 0

### **`pmu_element_size`**

PMU element size (0: 0KB, 1: 4KB, 2: 8KB).

Type: `uint8_t`

Default value: 0

### **`pmu_element_start`**

PMU element start.

Type: `uint8_t`

Default value: 0

### **`pmu_latency_mon`**

Determines if Latency Monitoring is included.

Type: `uint8_t`

Default value: 1

### **`pmu_monitors`**

Number of PMU monitors present.

Type: `uint8_t`

Default value: 16

### **`pmu_present`**

Configures whether a PMU is included in the monitor.

Type: `uint8_t`

Default value: 0

### **`pmu_snapshot`**

Determines if PMU Snapshot interface is present.

Type: `uint8_t`

Default value: 1

### **`pmu_trace`**

Determines if the ATB Output Trace is included.

Type: `uint8_t`

Default value: 0

### **`secure_addr_space`**

Secure Address Space support.

Type: `uint8_t`

Default value: 1

## 3.121 BP141\_TZMA

Defined in `LISA/BP141_TZMA.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### About BP141\_TZMA

BP141\_TZMA permits a single physical memory cell of up to 2 MB to be shared between a secure and non-secure storage area. The partitioning between these areas is flexible.

This component routes transactions according to the following:

- The memory region that they are attempting to access.
- The security mode of the transaction.

The BP141\_TZMA fixes the base address of the secure region to the base address of the decode space. It uses the `R0SIZE` [9:0] input to configure the size of the secure region in 4 KB increments up to a maximum of 2 MB.

`TZMEMSIZE` is the maximum addressing range of the memory as defined by that parameter. By default, `TZMEMSIZE` is set to 2 MB. In the following table, `AxADDR` is the offset address that the transactions want to access:

**Table 3-432: BP141\_TZMA security control**

AxADDR   Memory Region   Non-secure Transfer   Secure Transfer   ===== +=====+=====+=====+ AxADDR < R0Size   Secure, R0   Illegal   Legal
R0SIZE <= AxADDR and AxADDR < TZMEMSIZE   Non-secure, R1   Legal   Legal
AxADDR => TZMEMSIZE   No access   Illegal   Illegal

### Iris and MTI instances for BP141\_TZMA

This model has the following Iris instances:

Name	Instance type
BP141_TZMA	BP141_TZMA
BP141_TZMA.pvbusrange_0	PVBusRange
BP141_TZMA.pvbusrange_0.pvbus_mapper	PVBusMapper

Name	Instance type
BP141_TZMA.tzswitch_0	TZSwitch
BP141_TZMA.tzswitch_0.pvbus_mapper	PVBusMapper

This model has the following MTI trace components:

Name	Component type
BP141_TZMA.pvbusrange_0.pvbus_mapper	PVBusMapper
BP141_TZMA.tzswitch_0.pvbus_mapper	PVBusMapper

### Ports for BP141\_TZMA

Port	Direction	Protocol	Description
pv_output	master	PVBus	Routed PVBus output
pvbus	slave	PVBus	Bus slave interface.
R0Size	slave	Value	A software interface that is driven from the TrustZone Protection Controller (TZPC), setting the secure region size by bits[9:0].

### Parameters for BP141\_TZMA

#### TZMEMSIZE

Addressable range of device.

Type: uint32\_t

Default value: 0x200000

#### TZSECROMSIZE

Default secure size.

Type: uint32\_t

Default value: 0x200

#### TZSEGSIZE

Segment size.

Type: uint32\_t

Default value: 0x1000

## 3.122 BP147\_TZPC

Defined in LISA/BP147\_TZPC.lisa.

This model supports the following revisions of the IP at the given quality levels:

Copyright © 2017–2026 Arm Limited (or its affiliates). All rights reserved.  
Non-Confidential

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### About BP147\_TZPC

BP147\_TZPC provides a software interface to the protection bits in a secure system in a TrustZone design.

### Iris and MTI instances for BP147\_TZPC

This model has the following Iris instances:

Name	Instance type
BP147_TZPC	BP147_TZPC
BP147_TZPC.busslave	PVBusSlave

This model has the following MTI trace components:

Name	Component type
BP147_TZPC.busslave	PVBusSlave

### Ports for BP147\_TZPC

Port	Direction	Protocol	Description
bus_in_s	slave	PVBus	Slave port for register access.
TZPCDECPROT0	master	Value	Output decode protection 0 status.
TZPCDECPROT1	master	Value	Output decode protection 1 status.
TZPCDECPROT2	master	Value	Output decode protection 2 status.
TZPCROSIZE	master	Value	Output secure RAM region size.

### Parameters for BP147\_TZPC

This component does not have any parameters.

## 3.123 Base\_PowerController

Defined in `examples/LISA/Common/LISA/Base_PowerController.lisa`.

### About Base\_PowerController

The Base Platform Power Controller component provides a basic register interface for software to control the power up and power down of cores in the cluster.

Identify cores in the system to the Base\_PowerController by writing 24 bits in MPIDR format, providing the following levels of affinity:

**Bits [23:16]**

Affinity level 2.

**Bits [15:8]**

Affinity level 1.

**Bits [7:0]**

Affinity level 0.

Examples of affinity usage are `not_applicable/cluster/processor` and `cluster/processor/thread`.

To identify which cores to power up at startup, use parameter `pctl.startup`.

Specify core affinities with a dotted-quad. Wildcards are allowed. The format depends on the architecture:

- In Armv8.1 and earlier, use:

```
-C pctl.startup=0.0.Y.X
```

where X refers to the core number and Y refers to the cluster number. For example `0.0.0.0` refers to `cluster0.cpu0` and `0.0.1.1` refers to `cluster1.cpu1`. Use wildcards to indicate all cores at an affinity level. For example, to turn on all the cores in cluster 0, use `0.0.0.*`.

- In Armv8.2 and later, use:

```
-C pctl.startup=0.Z.Y.X
```

where X refers to the thread number, Y refers to the core number, and Z refers to the cluster number.

**Base\_PowerController registers**

The following table summarises the power control registers in order of offset from the base memory address.

After the table, there is more information about each register.

**Table 3-440: Base\_PowerController register summary**

Offset	Name	Type	Reset	Width	Description
0x00	PPOFFR	RW	0x---	32	Power Control Processor Off Register
0x04	PPONR	RW	0x---	32	Power Control Processor On Register
0x08	PCOFFR	RW	0x---	32	Power Control Cluster Off Register
0x0C	PWKUPR	RW	0x---	32	Power Control Wakeup Register
0x10	PSYSR	RW	0x---	32	Power Control SYS Status Register

**PPOFFR**

The Power Control Processor Off Register (PPOFFR) has the following characteristics:

**Purpose**

Processor SUSPEND command when PWKUPR and the GIC are programmed appropriately to provide wakeup events from IRQ and FIQ events to that processor.

**Usage constraints**

Processor must make power-off requests only for itself.

**Configurations**

Available in all configurations.

**Attributes**

See the register summary table.

**Table 3-441: Power Control Processor Off Register bit assignments**

Bits	Name	Function
[31:24]	-	Reserved
[23:0]	ID	MPIDR format affinity value of the processor to be switched off. Programming error if MPIDR != self.

**PPONR**

The Power Control Processor On Register (PPONR) has the following characteristics:

**Purpose**

Brings up a processor from low-power mode.

**Usage constraints**

Processor must make power-on requests only for other powered-off processors in the system.

**Configurations**

Available in all configurations.

**Attributes**

See the register summary table.

**Table 3-442: Power Control Processor On Register bit assignments**

Bits	Name	Function
[31:24]	-	Reserved
[23:0]	ID	MPIDR format affinity value of the processor to be switched on. Programming error if MPIDR == self.

**PCOFFR**

The Power Control Cluster Off Register (PCOFFR) has the following characteristics:

**Purpose**

Turns the cluster off.

**Usage constraints**

Cluster must make power-off requests only for itself.

**Configurations**

Available in all configurations.

**Attributes**

See the register summary table.

**Table 3-443: Power Control Cluster Off Register bit assignments**

Bits	Name	Function
[31:24]	-	Reserved
[23:0]	ID	MPIDR format affinity value of powered-on processor in the cluster to be switched off. Programming error if MPIDR != self.

**PWKUPR**

The Power Control Wakeup Register (PWKUPR) has the following characteristics:

**Purpose**

Configures whether wakeup requests from the GIC are enabled for this cluster.

**Usage constraints**

There are no usage constraints.

**Configurations**

Available in all configurations.

**Attributes**

See the register summary table.

**Table 3-444: Power Control Wakeup Register bit assignments**

Bits	Name	Function
[31]	WEN	If set, enables wakeup interrupts (return from SUSPEND) for this cluster.
[30:24]	-	Reserved
[23:0]	ID	MPIDR format affinity value of processor whose Wakeup Enable bit is to be configured.

**PSYSR**

The Power Control SYS Status Register (PSYSR) has the following characteristics:

**Purpose**

Provides information on the powered status of a given core. Software writes bits [23:0] for the required core and reads the value along with the associated status in bits [31:24].

**Usage constraints**

There are no usage constraints.

**Configurations**

Available in all configurations.

**Attributes**

See the register summary table.



**Table 3-445: Power Control SYS Status Register bit assignments**

Bits	Name	Function
[31]	L2	Read-only. A value of 1 indicates that affinity level 2 is active/on. If affinity level 2 is not implemented this bit is <b>RAZ</b> .
[30]	L1	Read-only. A value of 1 indicates that affinity level 1 is active/on. If affinity level 1 is not implemented this bit is <b>RAZ</b> .
[29]	L0	Read-only. A value of 1 indicates that affinity level 0 is active/on.
[28]	WEN	Read-only. A value of 1 indicates wakeup interrupts, return from SUSPEND, enabled for this processor. This is an alias of PWKUPR.WEN for this core.
[27]	PC	Read-only. A value of 1 indicates pending cluster off, the cluster enters low-power mode the next time it raises signal STANDBYWFI2.
[26]	PP	Read-only. A value of 1 indicates pending processor off, the processor enters low-power mode the next time it raises signal STANDBYWFI.
[25:24]	WK	Read-only. Indicates the reason for LEVEL0 power on:  <b>0b00</b> Cold power-on  <b>0b01</b> System reset pin  <b>0b10</b> Wake by PPONR  <b>0b11</b> Wake by GIC WakeRequest signal
[23:0]	ID	MPIDR format affinity value.

### Iris and MTI instances for Base\_PowerController

This model has the following Iris instances:

Name	Instance type
Base_PowerController	Base_PowerController
Base_PowerController.busslave	PVBusSlave
Base_PowerController.timer_ppu_transition	ClockTimerThread
Base_PowerController.timer_ppu_transition.timer	ClockTimerThread64
Base_PowerController.timer_ppu_transition.timer.thread	SchedulerThread
Base_PowerController.timer_ppu_transition.timer.thread_event	SchedulerThreadEvent
Base_PowerController.timer_reset	ClockTimerThread
Base_PowerController.timer_reset.timer	ClockTimerThread64
Base_PowerController.timer_reset.timer.thread	SchedulerThread
Base_PowerController.timer_reset.timer.thread_event	SchedulerThreadEvent
Base_PowerController.timer_sys_reset_request	ClockTimerThread
Base_PowerController.timer_sys_reset_request.timer	ClockTimerThread64
Base_PowerController.timer_sys_reset_request.timer.thread	SchedulerThread
Base_PowerController.timer_sys_reset_request.timer.thread_event	SchedulerThreadEvent
Base_PowerController.utility_busX (where X = 0-3)	PVBusMaster

This model has the following MTI trace components:

Name	Component type
Base_PowerController	Base_PowerController
Base_PowerController.busslave	PVBusSlave
Base_PowerController.utility_busX (where X = 0–3)	PVBusMaster

### Ports for Base\_PowerController

Port	Direction	Protocol	Description
cpuporeset	master	Signal	-
dbgnoopwrdown	slave	Signal	-
l2reset	master	Signal	-
pchannel_m	master	PChannel	-
pdbus_s	slave	PVBus	-
standbywfi	slave	Signal	-
standbywfi12	slave	Signal	-
system_reset_req	slave	Signal	-
system_reset	master	Signal	-
utility_bus_m	master	PVBus	-
wakerequest	slave	Signal	-

### Parameters for Base\_PowerController

#### Affinity-shifted

Whether core number is reflected in Affinity1 instead of Affinity0.

Type: `bool`

Default value: `false`

#### CPU-affinities

Definition of which cores are attached to the control pins, as a comma separated list of affinity dotted quads.

Type: `string`

Default value: `"0.0.0.0"`

#### CPU-available-mask

One bit per entry in CPU-affinities list, set zero if a CPU is wired up but actually not available.

Type: `uint64_t`

Default value: `0xffffffffffffffff`

**enable\_lock\_step**

If lock step is enabled, the number of available cores get reduced to half.

Type: `bool`

Default value: `false`

**startup**

Comma-separated list of cores (wildcards allowed) to be powered up at startup or system reset.

Type: `string`

Default value: `"0.0.0.*"`

**use\_in\_cluster\_ppu**

Set this to true if base power controller is connected to V9 core where in-cluster PPU is used, false, otherwise.

Type: `bool`

Default value: `false`

**use\_pchannel\_for\_threads**

Set this to true if the pchannel is connected to cpus with thread support.

Type: `bool`

Default value: `false`

3.124 BroadcastSignal2AMBAPVSignal

Defined in `examples/SystemCEExport/Bridges/BroadcastSignal2AMBAPVSignal.lisa`.

About BroadcastSignal2AMBAPVSignal

Broadcast signal to AMBAPVSignal coverter.

Iris and MTI instances for BroadcastSignal2AMBAPVSignal

This model has the following Iris instances:

Name	Instance type
BroadcastSignal2AMBAPVSignal	<a href="#">BroadcastSignal2AMBAPVSignal</a>

Ports for BroadcastSignal2AMBAPVSignal

Port	Direction	Protocol	Description
<code>amba_pv_signal_m</code>	master	<a href="#">AMBAPVSignal</a>	-

Port	Direction	Protocol	Description
amba_pv_signal_s	slave	AMBAPVSignal	-
b_signal	broadcast	Signal	-

### Parameters for BroadcastSignal2AMBAPVSignal

This component does not have any parameters.

## 3.125 CCI400

Defined in `LISA/CCI400.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### About CCI400

If you disable the `cache_state_modelled` parameter, this component has negligible performance impact. If you enable `cache_state_modelled`, it adds significant cost to throughput for coherent transactions.

This model implements the slave interface Shareable Override Register, which can be read and written, but it has no functionality.

### ACE limitation

AXI Coherency Extensions (ACE) are extensions to AXI4 that support system-level cache coherency between multiple clusters. The ACE cache models in the Arm Cortex-A15 and Cortex-A7, and the ACE support in the CCI-400 have the limitation that they only process one transaction at a time. Normally, the simulation processes each transaction to completion before allowing any master to generate another transaction.

However, in the following situation the simulation might fail. If a SystemC bus slave calls `wait()` while it is processing a transaction, this call might allow another master to issue another transaction that passes through the CCI-400 or the Cortex-A15 or Cortex-A7 caches. This situation could happen if a SystemC bus master running in another thread is connected to one of the ACE-lite ports on the CCI-400.

### Iris and MTI instances for CCI400

This model has the following Iris instances:

Name	Instance type
CCI400	CCI400
CCI400.cciinterconnect	PVCache

Name	Instance type
CCI400.cciregisters	CCIRegisters
CCI400.cciregisters.clocktimer	ClockTimerThread
CCI400.cciregisters.clocktimer.timer	ClockTimerThread64
CCI400.cciregisters.clocktimer.timer.thread	SchedulerThread
CCI400.cciregisters.clocktimer.timer.thread_event	SchedulerThreadEvent

This model has the following MTI trace components:

Name	Component type
CCI400	CCI400

### Ports for CCI400

Port	Direction	Protocol	Description
acchannelen	slave	Value	For each upstream port, determine if it is enabled or not with respect to snoop requests.
barrierterminate	slave	Value	For each downstream port, determine if barriers are terminated at that port.
broadcastcachemain	slave	Value	For each downstream port, determine if broadcast cache maintenance operations are forwarded down that port. A three bit signal but as the model only have a single downstream port, setting any of the bits will make it work.
bufferableoverride	slave	Value	For each downstream port, determine if all transactions are forced to non-bufferable (AWCACHE[0] is forced to 0).
clk_in	slave	ClockSignal	Clock signal for cciregisters
errorirq	master	Signal	A signal stating that the imprecise error register is nonzero.
evntcntoverflow	master	Signal	When an event counter overflows, it sets the corresponding signal.
lint_ace_3_reset_state	slave	Signal	This port can be connected to the reset signals of the system attached to the pvbus_s_ace_3 port.
lint_ace_4_reset_state	slave	Signal	This port can be connected to the reset signals of the system attached to the pvbus_s_ace_4 port.
periphbase	slave	Value_64	This port sets the base address of the private peripheral region.
pvbus_m	master	PVBus	Master port for all downstream memory accesses.
pvbus_s_ace_3	slave	PVBus	ACE-capable slave ports.
pvbus_s_ace_4	slave	PVBus	ACE-capable slave ports.
pvbus_s_ace_lite_plus_dvm_0	slave	PVBus	Memory bus interface that implements ACE lite and DVM protocol.
pvbus_s_ace_lite_plus_dvm_1	slave	PVBus	Memory bus interface that implements ACE lite and DVM protocol.
pvbus_s_ace_lite_plus_dvm_2	slave	PVBus	Memory bus interface that implements ACE lite and DVM protocol.
reset_in	slave	Signal	Signal to reset the CCI.
reset_state_of_ace_lite_ports	slave	Signal	This port can be connected to the reset signals of the system attached to ACE-Lite ports 0,1,2

## Parameters for CCI400

### **acchannelen**

For each upstream port, determine if it is enabled or not w.r.t. snoop requests.

Type: `uint32_t`

Default value: 31

### **barrierterminate**

For each downstream port, determine if barriers will be terminated at that port.

Type: `uint32_t`

Default value: 7

### **broadcastcachemain**

For each downstream port a bit determines if broadcast cache maintenance operations are forwarded down that port.

Type: `uint32_t`

Default value: 0

### **bufferableoverride**

For each downstream port, determine if all transactions will be forced to non-bufferable.

Type: `uint32_t`

Default value: 0

### **cache\_state\_modelled**

Model the cache state.

Type: `bool`

Default value: `true`

### **force\_on\_from\_start**

Enables snooping on upstream ports from the start of simulation.

The CCI will normally start up with snooping disabled, however, using this parameter we allow the model to start with snooping enabled without SW drivers programming the CCI.

This is only setup at simulation reset and not at signal reset.

If the upstreams can ever be held in reset then you *must* connect:

- `reset_state_of_ace_lite_ports[]`
- `lint_ace_3_reset_state`
- `lint_ace_4_reset_state`

so that it knows when to disable snoops to the upstream systems.

Otherwise, the upstream system will receive snoop messages whilst in reset and complain that it 'received a snoop request whilst it was in reset'.

Type: `bool`

Default value: `false`

### **`is_downstream_domain_boundary_for_far_atomic`**

This interconnect is at the last stage of the domain boundary.

Type: `bool`

Default value: `false`

### **`log_enabled`**

Enable log messages from the CCI register file.

**0**

do not print anything

**1**

print only access violations

**2**

also print writes

**3**

print reads as well.

Type: `int`

Default value: `1`

### **`periphbase`**

Value for PERIPHBASE. Only bits [39:16] are used. This value may be overridden by an input on the periphbase port.

Type: `uint64_t`

Default value: `0x2c000000`

### **`revision`**

Revision of the CCI400.

Type: `string`

Default value: `"r0p0"`

## 3.126 CCI500

Defined in `LISA/CCI500.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support
r0p2	Full support
r1p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### About CCI500

The LISA file declares seven upstream ports. You can configure these ports with `num_ace_ports` and `num_ace_lite_ports`:

- The bottom `num_ace_lite_ports` are ACE-Lite+DVM.
- The next `num_ace_ports` are ACE.
- Any remaining ports are ignored. If transactions are made on them, then warnings are produced.

For example, if `num_ace_ports = 1` and `num_ace_lite_ports = 1` then

- `pvbus_s[1]` is ACE
- `pvbus_s[0]` is ACE-Lite+DVM
- `pvbus_s[6-2]` are considered not to exist.

### Differences between the model and the RTL

#### Address Decoder

- Only supports striping down to 4KiB.
- If the address decoder aborts the access, returns `SLVERR` rather than `DECERR`.

#### Interfaces

- The model does not implement the Q-Channel and P-Channel interfaces.

#### Performance Monitoring Unit

- PMU counters recognize only a few event sources:
  - Slave interface events:



**3**

ReadOnce.

**4**

ReadClean, ReadShared, ReadNotSharedDirty, ReadUnique.

**5**

MakeUnique, CleanUnique.

**6**

CleanInvalid, CleanShared, MakeInvalid.

**7**

DVM transaction received from upstream.

**9**

Read data that is satisfied by a snoop request.

- No events are implemented for the global events or for the master events.
- The PMU does not implement the event bus (EVNTBUS).

**Register access**

- The register file only supports 32-bit accesses to its registers. Later versions of the CCI500 hardware support full write strobes to the register file. This limitation means that byte and halfword accesses work on the hardware but not on this version of the component.

**Snoop filter RAMs**

- Snoop filter RAMs are not modeled. The Status Register fields that relate to the power state of the Snoop filter RAMs are undefined.

**Iris and MTI instances for CCI500**

This model has the following Iris instances:

Name	Instance type
CCI500	CCI500
CCI500.downstream[Y].pvbusmaster (where Y = 0–5)	PVBusMaster
CCI500.pvbus_register_file_s[0]	PVBusSlave
CCI500.upstream[Y] (where Y = 0–6)	PVBusSlave

This model has the following MTI trace components:

Name	Component type
CCI500	CCI500
CCI500.downstream[Y].pvbusmaster (where Y = 0–5)	PVBusMaster
CCI500.pvbus_register_file_s[0]	PVBusSlave
CCI500.upstream[Y] (where Y = 0–6)	PVBusSlave

## Ports for CCI500

Port	Direction	Protocol	Description
acchannelensx	slave	Value	ACCHANNELENSx represents the ports ACCHANNELENS0..ACCHANNELENS7 on the RTL (assuming there are seven upstream ports). * each upstream ACE port 'y' (pvbus_s[y]) has a two bit ACCHANNELENSx * bit 0 == 0 – DVM messages are disabled from being sent to this interface * bit 1 == 0 – Snoop messages are disabled from being sent to this interface * each upstream ACE-Lite port 'z' (pvbus_s[z]) has a one bit ACCHANNELENSx * bit 0 == 0 – DVM messages are disabled from being sent to this interface In the model, as we support a variety of configurations with a single LISA file then each port will behave as though it is one bit or two bit as appropriate. If you send a value that cannot be represented, given the width of the port, then the CCI model will halt and produce a fatal error. The assumed values of these are set by parameters until they are driven, so you need not drive them if they are constant. In the RTL, these signals are sampled at reset. Due to ordering issues w.r.t. reset() on different components then we cannot do that. Instead the signals are sampled at first transaction. Thus any controller that is producing these signals has to hold them constant for long enough. AC channel enables.
address_decoder	master	CCI500_AddressDecoderProtocol	An address decoder can be attached to the address_decoder port to choose which pvbus_s port a downstream transaction will go out of. If you do not connect an address decoder then all transactions will go out of port 0.
dbgen	slave	Signal	Invasive debug enable.
errirq	master	Signal	Indicates that an error response, DECERR or SLVERR, is received on the RRESP, BRESP, or CRRESP input signals, and it cannot be signaled precisely.
evntcntoverflow	master	Signal	Overflow flags for the PMU clock and counters.
niden	slave	Signal	Non-invasive debug enable.
pvbus_m	master	PVBus	Bus master ports.
pvbus_register_file_s	slave	PVBus	The slave port of the register file.
pvbus_s	slave	PVBus	Bus slave ports.
reset_in	slave	Signal	Reset the interconnect.
reset_state_of_upstream_port	slave	Signal	Tell the interconnect the reset state of the upstream ports, this can be used by the interconnect to check some aspects of the reset sequencing. If you are using force_on_from_start then you <i>must</i> connect these pins.
spiden	slave	Signal	Secure invasive debug enable.
spniden	slave	Signal	Secure privileged non-invasive debug enable.

## Parameters for CCI500

### acchannelens0

For upstream port `pvbus_s[0]`, if `bit[0] == 0` then DVM messages are permanently disabled from being sent.

If this is an ACE port, then if `bit[1] == 0` then snoop messages are permanently disabled from being sent.

This parameter can be overridden by the signal `acchannelensx[0]`.

For an ACE-Lite port then `bit[1]` is ignored from the parameter, allowing the default value of `0x3` create a functional system without excessive configuration.

Type: `uint64_t`

Default value: `0x0`

### **acchannelens1**

For upstream port `pvbuss[1]`, if `bit[0] == 0` then DVM messages are permanently disabled from being sent.

If this is an ACE port, then if `bit[1] == 0` then snoop messages are permanently disabled from being sent.

This parameter can be overridden by the signal `acchannelensx[1]`.

For an ACE-Lite port then `bit[1]` is ignored from the parameter, allowing the default value of `0x3` create a functional system without excessive configuration.

Type: `uint64_t`

Default value: `0x0`

### **acchannelens2**

For upstream port `pvbuss[2]`, if `bit[0] == 0` then DVM messages are permanently disabled from being sent.

If this is an ACE port, then if `bit[1] == 0` then snoop messages are permanently disabled from being sent.

This parameter can be overridden by the signal `acchannelensx[2]`.

For an ACE-Lite port then `bit[1]` is ignored from the parameter, allowing the default value of `0x3` create a functional system without excessive configuration.

Type: `uint64_t`

Default value: `0x0`

### **acchannelens3**

For upstream port `pvbuss[3]`, if `bit[0] == 0` then DVM messages are permanently disabled from being sent.

If this is an ACE port, then if `bit[1] == 0` then snoop messages are permanently disabled from being sent.

This parameter can be overridden by the signal `acchannelensx[3]`.

For an ACE-Lite port then `bit[1]` is ignored from the parameter, allowing the default value of `0x3` create a functional system without excessive configuration.

Type: `uint64_t`

Default value: `0x0`

#### **acchannelens4**

For upstream port `pvbuss[4]`, if `bit[0] == 0` then DVM messages are permanently disabled from being sent.

If this is an ACE port, then if `bit[1] == 0` then snoop messages are permanently disabled from being sent.

This parameter can be overridden by the signal `acchannelensx[4]`.

For an ACE-Lite port then `bit[1]` is ignored from the parameter, allowing the default value of `0x3` create a functional system without excessive configuration.

Type: `uint64_t`

Default value: `0x0`

#### **acchannelens5**

For upstream port `pvbuss[5]`, if `bit[0] == 0` then DVM messages are permanently disabled from being sent.

If this is an ACE port, then if `bit[1] == 0` then snoop messages are permanently disabled from being sent.

This parameter can be overridden by the signal `acchannelensx[5]`.

For an ACE-Lite port then `bit[1]` is ignored from the parameter, allowing the default value of `0x3` create a functional system without excessive configuration.

Type: `uint64_t`

Default value: `0x0`

#### **acchannelens6**

For upstream port `pvbuss[6]`, if `bit[0] == 0` then DVM messages are permanently disabled from being sent.

If this is an ACE port, then if `bit[1] == 0` then snoop messages are permanently disabled from being sent.

This parameter can be overridden by the signal `acchannelensx[6]`.

For an ACE-Lite port then `bit[1]` is ignored from the parameter, allowing the default value of `0x3` create a functional system without excessive configuration.

Type: `uint64_t`

Default value: `0x0`

### **addr\_width**

The bit-width of the address that the CCI can accept.

Type: `uint64_t`

Default value: `40`

### **cache\_state\_modelled**

Model the cache state.

Type: `bool`

Default value: `1`

### **dbgen**

Invasive debug enable. If true, enables the counting of PMU events.

Type: `bool`

Default value: `true`

### **enable\_logger**

Enable PVBUSLoggers for the downstream ports in the CCI model.

Type: `bool`

Default value: `false`

### **force\_on\_from\_start**

Enables snooping on upstream ports from the start of simulation.

The interconnect will normally start up with snooping/DVM disabled. This parameter enables the upstream system of a port to be snooped if the upstream is not in reset and if `ACCHANNELENSx` allows it.

No software driver for the interconnect is needed.

Any port that could go into reset must have `reset_state_of_upstream_port[]` reflect the reset state of that upstream system.

Otherwise, the upstream system may receive snoop/DVM messages whilst in reset and may complain that it 'received a snoop request whilst it was in reset'.

Do not use if software is directly controlling the interconnect.

This option does not remove the responsibility of the upstream system to clean any shared dirty data from its caches before going into reset.

Type: `bool`

Default value: `false`

### **niden**

Whether non-secure events are allowed to be counted in the performance monitor.

Type: `bool`

Default value: `true`

### **num\_ace\_lite\_ports**

The number of ACE-Lite+DVM ports. These are the lowest numbered ports. The total number of ports must not exceed 7.

Type: `unsigned`

Default value: 5

### **num\_ace\_ports**

The top `num_ace_ports` are ACE and support full coherency. The total number of ports must not exceed 7.

Type: `unsigned`

Default value: 2

### **number\_of\_phantom\_entries**

"Number of phantom entries in the cache. Phantom entries are used by certain cache operations to hold temporary data. Usually this should be left at the default value which is safe for all systems containing up to 32 masters.

Type: `uint64_t`

Default value: `0x0000000000000020`

**qos\_threshold\_upper**

Reset value for the QoS threshold register.

Type: `uint64_t`

Default value: `0x0000000000000000c`

**reentrancy\_support**

Must be one of:

**on**

hazard checking per cache line (normal mode)

**off**

no hazard checking (use only for single master systems)

**cacheglobal**

hazard checking globally for cache (not per cache line, testing feature, provokes more hazards than necessary)

**env**

take value from `FM_REENTRANCY_SUPPORT` env var, if this is not set use 'on'.

Type: `string`

Default value: `"env"`

**spiden**

Secure invasive debug enable. If both `SPIDEN` and `DBGEN` are high, enables the counting of both Non-secure and Secure events.

Type: `bool`

Default value: `true`

**spniden**

Whether secure and non-secure events are allowed to be counted in the performance monitor.

Type: `bool`

Default value: `true`

**version**

The version of the interconnect.

Allowed versions are:

- `rOp0`
- `rOp2`

- r1p0.

Type: `string`

Default value: `""`

## 3.127 CCI550

Defined in `LISA/CCI550.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support
r1p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Differences between the model and the RTL

#### Address Decoder

- Only supports striping down to 4KiB.
- If the address decoder aborts the access, returns SLVERR rather than DECERR.

#### Interfaces

- The model does not implement the Q-Channel and P-Channel interfaces.

#### Performance Monitoring Unit

- PMU counters recognize only a few event sources:
  - Slave interface events:
    - 3** ReadOnce.
    - 4** ReadClean, ReadShared, ReadNotSharedDirty, ReadUnique.
    - 5** MakeUnique, CleanUnique.
    - 6** CleanInvalid, CleanShared, MakeInvalid.
    - 7** DVM transaction received from upstream.
    - 9** Read data that is satisfied by a snoop request.



- No events are implemented for the global events or for the master events.
- The PMU does not implement the event bus (EVNTBUS).

### Reset signal sampling

- The configuration ports `acchannelensx[]` are sampled in the hardware when coming out of reset. In the model, these ports are sampled at the first transaction to a `pvbus_s` port or to the register file.

### Status Register, change-pending, and DVM messages

- The Status Register provides information on when the last transaction that could have observed an old value of a snoop or DVM enable has finished in the upstream system. Therefore a port that has been disabled can now have the system upstream of that port turned off. The model does not track DVM messages in the upstream system.

### Snoop filter RAMs

- The CCI-550 hardware has a snoop filter that reduces the number of snoop requests that the interconnect has to make. The model does not have a snoop filter and could make more snoop requests than the hardware would. This difference has no programmer-visible effect.
- The Status Register fields that relate to the power state of the Snoop filter RAMs are undefined.

### Registers

- The following registers provide storage but have no effect on the model.
  - QoS registers.
  - Interface monitor registers. These registers are intended for silicon debug.

### Iris and MTI instances for CCI550

This model has the following Iris instances:

Name	Instance type
CCI550	CCI550
CCI550.downstream[Y].pvbusmaster (where Y = 0–6)	PVBusMaster
CCI550.pvbus_register_file_s[0]	PVBusSlave
CCI550.upstream[Y] (where Y = 0–6)	PVBusSlave

This model has the following MTI trace components:

Name	Component type
CCI550	CCI550
CCI550.downstream[Y].pvbusmaster (where Y = 0–6)	PVBusMaster
CCI550.pvbus_register_file_s[0]	PVBusSlave
CCI550.upstream[Y] (where Y = 0–6)	PVBusSlave

## Ports for CCI550

Port	Direction	Protocol	Description
acchannelensx	slave	Value	The acchannelensx[N] pins are used to tell the interconnect if the upstream system will accept snoops and/or DVM messages.
address_decoder	master	CCI500_AddressDecoderProtocol	An address decoder can be attached to the address_decoder port to choose which pvbuss port a downstream transaction will go out of. If you do not connect an address decoder then all transactions will go out of port 0.
dbgen	slave	Signal	Invasive debug enable.
errirq	master	Signal	Some async error was detected.
evntcntoverflow	master	Signal	The output interrupts of the event counters.
niden	slave	Signal	Non-invasive debug enable.
pvbus_m	master	PVBus	The downstream master ports.
pvbus_register_file_s	slave	PVBus	The slave port of the register file.
pvbus_s	slave	PVBus	Bus slave ports.
reset_in	slave	Signal	Reset the interconnect.
reset_state_of_upstream_port	slave	Signal	Tell the interconnect the reset state of the upstream ports, this can be used by the interconnect to check some aspects of the reset sequencing. If you are using force_on_from_start then you <i>must</i> connect these pins.
sci_s	slave	SystemCoherencyInterface	The System Coherency Interface bus. For those upstream ports that have a corresponding bit set in the bitmap of si_system_coherency_interface then the corresponding sci_m port can be used to move the upstream system into and out of the coherency domain.
spiden	slave	Signal	Secure privileged invasive debug enable.
spniden	slave	Signal	Secure privileged non-invasive debug enable.

## Parameters for CCI550

### acchannelens0

For upstream port 0 determine if it is enabled or not w.r.t. snoop requests.

Type: uint64\_t

Default value: 0x0000000000000000

### acchannelens1

For upstream port 1 determine if it is enabled or not w.r.t. snoop requests.

Type: uint64\_t

Default value: 0x0000000000000000

### acchannelens2

For upstream port 2 determine if it is enabled or not w.r.t. snoop requests.

Type: `uint64_t`

Default value: `0x0000000000000000`

### **acchannelens3**

For upstream port 3 determine if it is enabled or not w.r.t. snoop requests.

Type: `uint64_t`

Default value: `0x0000000000000000`

### **acchannelens4**

For upstream port 4 determine if it is enabled or not w.r.t. snoop requests.

Type: `uint64_t`

Default value: `0x0000000000000000`

### **acchannelens5**

For upstream port 5 determine if it is enabled or not w.r.t. snoop requests.

Type: `uint64_t`

Default value: `0x0000000000000000`

### **acchannelens6**

For upstream port 6 determine if it is enabled or not w.r.t. snoop requests.

Type: `uint64_t`

Default value: `0x0000000000000000`

### **addr\_width**

The bit-width of the address that the CCI can accept.

Type: `uint64_t`

Default value: `0x0000000000000028`

### **cache\_state\_modelled**

Model the cache state.

Type: `bool`

Default value: `true`

### **dbgen**

Invasive debug enable. If true, enables the counting of PMU events.

Type: `bool`

Default value: `true`

### **enable\_logger**

Enable PVBUSLoggers for the downstream ports in the CCI model.

Type: `bool`

Default value: `false`

### **force\_on\_from\_start**

The interconnect will normally start up with snooping/DVM disabled.

The parameter `si_system_coherency_interface` determines which connections are managed by the System Coherency Interface (SCI).

For connections that are managed by SCI, then this parameter has no effect.

For all other connections, this parameter enables the upstream system of a port to be snooped if the upstream is not in reset and if `ACCHANNLENSx` allows it.

No software driver for the interconnect is needed.

Any non-SCI port that could go into reset must have `reset_state_of_upstream_port[]` reflect the reset state of that upstream.

Otherwise, the upstream system may receive snoop/DVM messages whilst in reset and may complain that it 'received a snoop request whilst it was in reset'.

Do not use if software is directly controlling the interconnect. This option does not disavow responsibility of the upstream system to clean any shared dirty data from its caches before going into reset.

Type: `bool`

Default value: `false`

### **niden**

Whether non-secure events are allowed to be counted in the performance monitor.

Type: `bool`

Default value: `true`

### **num\_ace\_lite\_ports**

The bottom `num_ace_lite_ports` are ACE-Lite+DVM.

Type: `uint64_t`

Default value: `0x0000000000000005`

### **num\_ace\_ports**

The top `num_ace_ports` are ACE and support full coherency.

Type: `uint64_t`

Default value: `0x0000000000000002`

### **number\_of\_phantom\_entries**

Number of phantom entries in the cache. Phantom entries are used by certain cache operations to hold temporary data. Usually this should be left at the default value which is safe for all systems containing up to 32 masters.

Type: `uint64_t`

Default value: `0x0000000000000020`

### **qos\_threshold\_upper**

Reset value for the QoS threshold register.

Type: `uint64_t`

Default value: `0x000000000000000c`

### **reentrancy\_support**

Must be one of:

#### **on**

hazard checking per cache line (normal mode)

#### **off**

no hazard checking (use only for single master systems)

#### **cacheglobal**

hazard checking globally for cache (not per cache line, testing feature, provokes more hazards than necessary)

#### **env**

take value from `FM_REENTRANCY_SUPPORT` env var, if this is not set use 'on'.

Type: `string`

Default value: `"env"`

**si0\_qos\_bw\_regulator**

For upstream port 0 determine if it has a BW regulator. The effect of QoS is not modelled and this parameter only alters some registers.

Type: `bool`

Default value: `false`

**si1\_qos\_bw\_regulator**

For upstream port 1 determine if it has a BW regulator. The effect of QoS is not modelled and this parameter only alters some registers.

Type: `bool`

Default value: `false`

**si2\_qos\_bw\_regulator**

For upstream port 2 determine if it has a BW regulator. The effect of QoS is not modelled and this parameter only alters some registers.

Type: `bool`

Default value: `false`

**si3\_qos\_bw\_regulator**

For upstream port 3 determine if it has a BW regulator. The effect of QoS is not modelled and this parameter only alters some registers.

Type: `bool`

Default value: `false`

**si4\_qos\_bw\_regulator**

For upstream port 4 determine if it has a BW regulator. The effect of QoS is not modelled and this parameter only alters some registers.

Type: `bool`

Default value: `false`

**si5\_qos\_bw\_regulator**

For upstream port 5 determine if it has a BW regulator. The effect of QoS is not modelled and this parameter only alters some registers.

Type: `bool`

Default value: `false`

**si6\_qos\_bw\_regulator**

For upstream port 6 determine if it has a BW regulator. The effect of QoS is not modelled and this parameter only alters some registers.

Type: `bool`

Default value: `false`

**si\_system\_coherency\_interface**

This parameter tells the interconnect which upstream ports should be controlled by the System Coherency Interface.

Each bit corresponds to an upstream port, bit 0 to upstream port 0, etc.

If the SCI port is connected but `si_system_coherency_interface` disables its use then messages from the upstream will be ignored and software must manage the upstream system's entrance and exit of the coherency domain.

Type: `uint64_t`

Default value: `0x0000000000000000`

**spiden**

Secure invasive debug enable.

If both `SPIDEN` and `DBGEN` are high, enables the counting of both Non-secure and Secure events.

Type: `bool`

Default value: `true`

**spniden**

Whether secure and non-secure events are allowed to be counted in the performance monitor.

Type: `bool`

Default value: `true`

**version**

The version of the interconnect. Allowed versions are:

- `r0p0`
- `r1p0`.

Type: `string`

Default value: `""`

## 3.128 CCN502

Defined in `LISA/CCN502.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### About CCN502

CCN502 can be used for connecting components using the ACE and ACE-Lite interfaces. It has an L3 cache that can provide coherency between up to four fully-coherent ACE clusters and nine I/O coherent masters. It can connect up to two memory elements to drive transaction requests. This interconnect can be configured to support six or eight crosspoints, both of which are implemented in the model.

CCN502 has three or five downstream ports, depending on the number of crosspoints:

- Two or four SN-F ports for the memory controller
- One Acelite port (HNI)

The CCN502 parameters are not exposed through Iris, but can be seen in `ccn502.lisa`. Some useful parameters are:

- `cache_state_modelled`
- `cache_size_in_kbytes`
- `systemaddrmap`
- `variant_name`

CCN-502 supports up to 4 SN-Fs. To enable 4 SN-Fs in the model, set the `variant_name` parameter to `CCN502_8XP`, which means the 8XP/4HNF configuration.

### Limitations

The model has the following limitations:

- No support for 3 SN striping.
- If there are multiple SN-Fs, the distribution of addresses to each SN-F is not guaranteed to match the hardware.
- The parameters for the CCNCache subcomponent are not accessible in System Canvas.

### Iris and MTI instances for CCN502

This model has the following Iris instances:



Name	Instance type
CCN502	CCN5XX
CCN502.bus_slave_ocm	PVBusSlave
CCN502.ccn502_hni_exclusive_monitor_0	PVBusExclusiveMonitor
CCN502.ccn502_hni_exclusive_monitor_0.bus_mapper	PVBusMapper
CCN502.ccn_cache	CCNCache
CCN502.ccn_cache.downstream[Y].pvbusmaster (where Y = 0–47)	PVBusMaster
CCN502.ccn_cache.upstream[Y] (where Y = 0–47)	PVBusSlave
CCN502.ccn_registers	CCNRegisterSet
CCN502.ccn_registers.bus_slave	PVBusSlave
CCN502.ccn_router	PVBusMapper

This model has the following MTI trace components:

Name	Component type
CCN502.bus_slave_ocm	PVBusSlave
CCN502.ccn502_hni_exclusive_monitor_0	PVBusExclusiveMonitor
CCN502.ccn502_hni_exclusive_monitor_0.bus_mapper	PVBusMapper
CCN502.ccn_cache	CCNCache
CCN502.ccn_cache.downstream[Y].pvbusmaster (where Y = 0–47)	PVBusMaster
CCN502.ccn_cache.upstream[Y] (where Y = 0–47)	PVBusSlave
CCN502.ccn_registers	CCNRegisterSet
CCN502.ccn_registers.bus_slave	PVBusSlave
CCN502.ccn_router	PVBusMapper

## Ports for CCN502

Port	Direction	Protocol	Description
pvbus_m_hni	master	PVBus	HNI downstream port.
pvbus_m_snf	master	PVBus	SNF downstream ports.
pvbus_s_rnf	slave	PVBus	RNF upstream ports.
pvbus_s_rni	slave	PVBus	RNI upstream ports
reset_in	slave	Signal	Reset signal.

## Parameters for CCN502

### **acchannelen\_rnf**

Bitmap for each rnf upstream port to determine if it is enabled or not w.r.t. snoop requests.

Type: uint32\_t

Default value: 15

**acchannelen\_rni**

Bitmap for each rni upstream port to determine if it is enabled or not w.r.t. dvm requests.

Type: uint32\_t

Default value: 0x1ff

**cache\_size\_in\_kbytes**

Size of the L3 cache to Model.

Type: uint32\_t

Default value: 4096

**cache\_state\_modelled**

Model the cache state.

Type: bool

Default value: true

**enable\_logger**

Enable PVBUSLoggers for the downstream ports in the CCN model.

Type: bool

Default value: false

**force\_on\_from\_start**

Enables snooping on upstream ports from the start of simulation.

The CCN502 will normally start up with snooping disabled. However, using this parameter allows the model to start with snooping enabled without having to program it. This is only setup at simulation reset and not at signal reset.

Type: bool

Default value: false

**number\_of\_snf**

Number of SNF nodes present.

Type: uint32\_t

Default value: 2

**periphbase**  
Value for PERIPHBASE. Only bits [43:24] are used.

Type: `uint64_t`

Default value: `0x2c000000`

**sbsx\_bridge\_present**  
value for sbsx bridge presence.

Type: `bool`

Default value: `true`

**systemaddrmap**  
Bitmap for 20 regions in CCN Interconnect. Every two bits describes the region type for corresponding region.

Type: `uint64_t`

Default value: `0`

**variant\_name**  
Can be either CCN502\_6XP or CCN502\_8XP.

Type: `string`

Default value: `"CCN502_6XP"`

### 3.129 CCN504

Defined in `LISA/CCN504.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

#### About CCN504

CCN504 has three downstream ports:

- Two SNF ports for the memory controller
- One Acelite port (HNI)



The parameters for the `ccnCache` subcomponent are not accessible in System Canvas.

## Iris and MTI instances for CCN504

This model has the following Iris instances:

Name	Instance type
CCN504	CCN5XX
CCN504.ccn502_hni_exclusive_monitor_0	PVBusExclusiveMonitor
CCN504.ccn502_hni_exclusive_monitor_0.bus_mapper	PVBusMapper
CCN504.ccn_cache	CCNCache
CCN504.ccn_cache.downstream[Y].pvbusmaster (where Y = 0–47)	PVBusMaster
CCN504.ccn_cache.upstream[Y] (where Y = 0–47)	PVBusSlave
CCN504.ccn_registers	CCNRegisterSet
CCN504.ccn_registers.bus_slave	PVBusSlave
CCN504.ccn_router	PVBusMapper

This model has the following MTI trace components:

Name	Component type
CCN504.ccn502_hni_exclusive_monitor_0	PVBusExclusiveMonitor
CCN504.ccn502_hni_exclusive_monitor_0.bus_mapper	PVBusMapper
CCN504.ccn_cache	CCNCache
CCN504.ccn_cache.downstream[Y].pvbusmaster (where Y = 0–47)	PVBusMaster
CCN504.ccn_cache.upstream[Y] (where Y = 0–47)	PVBusSlave
CCN504.ccn_registers	CCNRegisterSet
CCN504.ccn_registers.bus_slave	PVBusSlave
CCN504.ccn_router	PVBusMapper

## Ports for CCN504

Port	Direction	Protocol	Description
pvbus_m_hni	master	PVBus	HNI downstream port.
pvbus_m_snf	master	PVBus	SNF downstream ports.
pvbus_s_rnf	slave	PVBus	RNF upstream ports.
pvbus_s_rni	slave	PVBus	RNI upstream ports.
reset_in	slave	Signal	Reset signal.

## Parameters for CCN504

### `acchannelen_rnf`

Bitmap for each rnf upstream port to determine if it is enabled or not w.r.t. snoop requests.

Type: `uint32_t`

Default value: 15

### **acchannelen\_rni**

Bitmap for each rni upstream port to determine if it is enabled or not w.r.t. dvm requests.

Type: `uint32_t`

Default value: `0x3ffff`

### **cache\_size\_in\_mbytes**

Size of the L3 cache to Model.

Type: `uint32_t`

Default value: 8

### **cache\_state\_modelled**

Model the cache state.

Type: `bool`

Default value: `true`

### **disable\_hni\_cacheable\_error**

When set to true, disables the generation of error response when HNI receives cacheable access.

Type: `bool`

Default value: `false`

### **enable\_logger**

Enable PVBUSLoggers for the downstream ports in the CCN model.

Type: `bool`

Default value: `false`

### **force\_on\_from\_start**

Enables snooping on upstream ports from the start of simulation.

The CCN504 will normally start up with snooping disabled. However, using this parameter allows the model to start with snooping enabled without having to program it. This is only setup at simulation reset and not at signal reset.

Type: `bool`

Default value: `false`

**number\_of\_snf**

Number of SNF nodes present.

Type: `uint32_t`

Default value: 2

**periphbase**

Value for PERIPHBASE. Only bits [43:24] are used.

Type: `uint64_t`

Default value: `0x2c000000`

**sbas\_bridge\_present**

value for sbas bridge presence.

Type: `bool`

Default value: `true`

**sbsx\_bridge\_present**

value for sbsx bridge presence.

Type: `bool`

Default value: `true`

**systemaddrmap**

Bitmap for 20 regions in CCN Interconnect. Every two bits describes the region type for corresponding region.

Type: `uint64_t`

Default value: 0

3.130 CCN508

Defined in `LISA/CCN508.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

## About CCN508

CCN508 has six downstream ports:

- Four SNF ports for the memory controller.
- Two Acelite ports (HNI).



Note

The parameters for the CCNCache subcomponent are not accessible in System Canvas.

## Iris and MTI instances for CCN508

This model has the following Iris instances:

Name	Instance type
CCN508	CCN5XX
CCN508.ccn502_hni_exclusive_monitor_z (where Z = 0-1)	PVBusExclusiveMonitor
CCN508.ccn502_hni_exclusive_monitor_z.bus_mapper (where Z = 0-1)	PVBusMapper
CCN508.ccn_cache	CCNCache
CCN508.ccn_cache.downstream[Y].pvbusmaster (where Y = 0-47)	PVBusMaster
CCN508.ccn_cache.upstream[Y] (where Y = 0-47)	PVBusSlave
CCN508.ccn_registers	CCNRegisterSet
CCN508.ccn_registers.bus_slave	PVBusSlave
CCN508.ccn_router	PVBusMapper

This model has the following MTI trace components:

Name	Component type
CCN508.ccn502_hni_exclusive_monitor_z (where Z = 0-1)	PVBusExclusiveMonitor
CCN508.ccn502_hni_exclusive_monitor_z.bus_mapper (where Z = 0-1)	PVBusMapper
CCN508.ccn_cache	CCNCache
CCN508.ccn_cache.downstream[Y].pvbusmaster (where Y = 0-47)	PVBusMaster
CCN508.ccn_cache.upstream[Y] (where Y = 0-47)	PVBusSlave
CCN508.ccn_registers	CCNRegisterSet
CCN508.ccn_registers.bus_slave	PVBusSlave
CCN508.ccn_router	PVBusMapper

## Ports for CCN508

Port	Direction	Protocol	Description
pvbus_m_hni	master	PVBus	HNI downstream ports.
pvbus_m_snf	master	PVBus	SNF downstream ports.

Port	Direction	Protocol	Description
pvbuss_s_rnf	slave	PVBus	RNF upstream ports.
pvbuss_s_rni	slave	PVBus	RNI upstream ports.
reset_in	slave	Signal	Reset signal.

## Parameters for CCN508

### **acchannelen\_rnf**

Bitmap for each rnf upstream port to determine if it is enabled or not w.r.t. snoop requests.

Type: `uint32_t`

Default value: 255

### **acchannelen\_rni**

Bitmap for each rni upstream port to determine if it is enabled or not w.r.t. dvm requests.

Type: `uint32_t`

Default value: 0xffffffff

### **cache\_size\_in\_mbytes**

Size of the L3 cache to Model.

Type: `uint32_t`

Default value: 8

### **cache\_state\_modelled**

Model the cache state.

Type: `bool`

Default value: `true`

### **enable\_logger**

Enable PVBusLoggers for the downstream ports in the CCN model.

Type: `bool`

Default value: `false`

### **force\_on\_from\_start**

Enables snooping on upstream ports from the start of simulation.



The CCN508 will normally start up with snooping disabled. However, using this parameter allows the model to start with snooping enabled without having to program it. This is only setup at simulation reset and not at signal reset.

Type: `bool`

Default value: `false`

### **number\_of\_snf**

Number of SNF nodes present.

Type: `uint32_t`

Default value: 2

### **periphbase**

Value for PERIPHBASE. Only bits [43:24] are used.

Type: `uint64_t`

Default value: `0x2c000000`

### **sbas\_bridge\_present**

value for sbas bridge presence.

Type: `bool`

Default value: `true`

### **sbsx\_bridge\_present**

value for sbsx bridge presence.

Type: `bool`

Default value: `true`

### **systemaddrmap**

Bitmap for 20 regions in CCN Interconnect. Every two bits describes the region type for corresponding region.

Type: `uint64_t`

Default value: 0

## 3.131 CCN512

Defined in `LISA/CCN512.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### About CCN512

CCN512 has an L3 cache that can provide coherency between up to 12 fully-coherent ACE clusters and 24 I/O coherent masters. It can connect up to four memory elements to drive transaction requests.

CCN512 has six downstream ports:

- Four SNF ports for the memory controller
- Two Acelite ports (HNI)

Some useful parameters are:

- `cache_state_modelled`
- `cache_size_in_mbytes`
- `systemaddrmap`

### Limitations

The model has the following limitations:

- No support for 3 SN striping.
- If there are multiple SN-Fs, the distribution of addresses to each SN-F is not guaranteed to match the hardware.
- The parameters for the CCNCache subcomponent are not accessible in System Canvas.

### Iris and MTI instances for CCN512

This model has the following Iris instances:

Name	Instance type
CCN512	CCN5XX
CCN512.bus_slave_ocm	PVBusSlave
CCN512.ccn502_hni_exclusive_monitor_Z (where Z = 0-1)	PVBusExclusiveMonitor
CCN512.ccn502_hni_exclusive_monitor_Z.bus_mapper (where Z = 0-1)	PVBusMapper
CCN512.ccn_cache	CCNCache
CCN512.ccn_cache.downstream[Y].pvbusmaster (where Y = 0-47)	PVBusMaster
CCN512.ccn_cache.upstream[Y] (where Y = 0-47)	PVBusSlave

Name	Instance type
CCN512.ccn_registers	CCNRegisterSet
CCN512.ccn_registers.bus_slave	PVBusSlave
CCN512.ccn_router	PVBusMapper

This model has the following MTI trace components:

Name	Component type
CCN512.bus_slave_ocm	PVBusSlave
CCN512.ccn502_hni_exclusive_monitor_Z (where Z = 0-1)	PVBusExclusiveMonitor
CCN512.ccn502_hni_exclusive_monitor_Z.bus_mapper (where Z = 0-1)	PVBusMapper
CCN512.ccn_cache	CCNCache
CCN512.ccn_cache.downstream[Y].pvbusmaster (where Y = 0-47)	PVBusMaster
CCN512.ccn_cache.upstream[Y] (where Y = 0-47)	PVBusSlave
CCN512.ccn_registers	CCNRegisterSet
CCN512.ccn_registers.bus_slave	PVBusSlave
CCN512.ccn_router	PVBusMapper

## Ports for CCN512

Port	Direction	Protocol	Description
pvbus_m_hni	master	PVBus	HNI downstream ports.
pvbus_m_snf	master	PVBus	SNF downstream ports.
pvbus_s_rnf	slave	PVBus	RNF upstream ports.
pvbus_s_rni	slave	PVBus	RNI upstream ports.
reset_in	slave	Signal	Reset signal.

## Parameters for CCN512

### **acchannelen\_rnf**

Bitmap for each rnf upstream port to determine if it is enabled or not w.r.t. snoop requests.

Type: uint32\_t

Default value: 0xffff

### **acchannelen\_rni**

Bitmap for each rni upstream port to determine if it is enabled or not w.r.t. dvm requests.

Type: uint32\_t

Default value: 0xffffffff

### **cache\_size\_in\_mbytes**

Size of the L3 cache to Model.

Type: `uint32_t`

Default value: 8

### **cache\_state\_modelled**

Model the cache state.

Type: `bool`

Default value: `true`

### **enable\_logger**

Enable PVBUSLoggers for the downstream ports in the CCN model.

Type: `bool`

Default value: `false`

### **force\_on\_from\_start**

Enables snooping on upstream ports from the start of simulation.

The CCN512 will normally start up with snooping disabled. However, using this parameter allows the model to start with snooping enabled without having to program it. This is only setup at simulation reset and not at signal reset.

Type: `bool`

Default value: `false`

### **number\_of\_snf**

Number of SNF nodes present.

Type: `uint32_t`

Default value: 2

### **periphbase**

Value for PERIPHBASE. Only bits [43:24] are used.

Type: `uint64_t`

Default value: `0x2c000000`

### **sbsx\_bridge\_present**

value for sbsx bridge presence.

Type: `bool`

Default value: `true`

### **systemaddrmap**

Bitmap for 20 regions in CCN Interconnect. Every two bits describes the region type for corresponding region.

Type: `uint64_t`

Default value: 0

## 3.132 CCSM\_F1

Defined in `LISA/CCSM_F1.lisa`.

### About CCSM\_F1

Clock Control State Machine F1.

### Iris and MTI instances for CCSM\_F1

This model has the following Iris instances:

Name	Instance type
CCSM_F1	CCSM_F1
CCSM_F1.CFMM	CFMM
CCSM_F1.CFMM.CFM	CFM
CCSM_F1.CFMM.CFM.CfmMaskOrGate	WideOrGate
CCSM_F1.CFMM.CFM.TrigOrGate	WideOrGate
CCSM_F1.CFMM.CFM.clksel_mux	Value_Multiplexer
CCSM_F1.CFMM.CFM.clock_mux	Clock_Multiplexer
CCSM_F1.CFMM.CFM.fw_clksel_mux	Value_Multiplexer
CCSM_F1.CFMM.CFM.trig_droop_mux	Signal_Multiplexer
CCSM_F1.CFMM.CFM.trig_soff_mux	Signal_Multiplexer
CCSM_F1.CFMM.CMM	CMM
CCSM_F1.CFMM.CMM.throttler0	Throttler
CCSM_F1.DVFSM	DVFSM
CCSM_F1.DVFSM.clock_muxY (where Y = 0-1)	Clock_Multiplexer

### Ports for CCSM\_F1

Port	Direction	Protocol	Description
<code>ccsm_ctrl_out</code>	master	Signal	-
<code>ccsm_data_in</code>	slave	ValueState	-
<code>ccsm_dynamic_out</code>	master	ValueState	-
<code>ccsm_event_in</code>	slave	Signal	-

Port	Direction	Protocol	Description
ccsm_glcm_sel	master	Value	-
ccsm_irq	master	Signal	-
ccsm_static_out	master	ValueState	-
clkin_pll0	slave	ClockSignal	-
clkin_pll1	slave	ClockSignal	-
clkout_dd	master	ClockSignal	-
clkout_fm	master	ClockSignal	-
clkout_mm	master	ClockSignal	-
pll0_bypass_en	master	Signal	-
pll0_dac_en	master	Signal	-
pll0_dsm_en	master	Signal	-
pll0_dynamic_en	master	Signal	-
pll0_dynamic	master	ValueState	-
pll0_en	master	Signal	-
pll0_fbdiv	master	ValueState	-
pll0_fj_done	slave	Signal	-
pll0_fj_frac_accuracy	master	ValueState	-
pll0_fj_freq_ch_slope	master	ValueState	-
pll0_fj_freq_ch_tau	master	ValueState	-
pll0_fj_start	master	Signal	-
pll0_foutpostdiv_en	master	Signal	-
pll0_foutvco_en	master	Signal	-
pll0_glcm_sel	master	Value	-
pll0_lock	slave	Signal	-
pll0_offsetcal_en	master	Signal	-
pll0_offsetcalbyp	master	Signal	-
pll0_offsetcalcnt	master	ValueState	-
pll0_offsetcalin	master	ValueState	-
pll0_offsetfastcal	master	Signal	-
pll0_postdiv1	master	ValueState	-
pll0_postdiv2	master	ValueState	-
pll0_refdiv	master	ValueState	-
pll0_rsvd	master	ValueState	-
pll0_static	master	ValueState	-
pll0_status	slave	ValueState	-
pll1_bypass_en	master	Signal	-
pll1_dac_en	master	Signal	-
pll1_dsm_en	master	Signal	-
pll1_dynamic_en	master	Signal	-
pll1_dynamic	master	ValueState	-

Port	Direction	Protocol	Description
pll1_en	master	Signal	-
pll1_fbdiv	master	ValueState	-
pll1_fj_done	slave	Signal	-
pll1_fj_frac_accuracy	master	ValueState	-
pll1_fj_freq_ch_slope	master	ValueState	-
pll1_fj_freq_ch_tau	master	ValueState	-
pll1_fj_start	master	Signal	-
pll1_foutpostdiv_en	master	Signal	-
pll1_foutvco_en	master	Signal	-
pll1_glcm_sel	master	Value	-
pll1_lock	slave	Signal	-
pll1_offsetcal_en	master	Signal	-
pll1_offsetcalbyp	master	Signal	-
pll1_offsetcalcnt	master	ValueState	-
pll1_offsetcalin	master	ValueState	-
pll1_offsetfastcal	master	Signal	-
pll1_postdiv1	master	ValueState	-
pll1_postdiv2	master	ValueState	-
pll1_refdiv	master	ValueState	-
pll1_rsvd	master	ValueState	-
pll1_static	master	ValueState	-
pll1_status	slave	ValueState	-
PORESETn	slave	Signal	-
reg_pvbus_s	slave	PVBus	-
RESETn	slave	Signal	-
spare_in	slave	Signal	-
spare_out	master	Signal	-
sysclk_in	slave	ClockSignal	-
trig_droop_alt	slave	Signal	-
trig_droop	slave	Signal	-
trig_overshoot	slave	Signal	-
trig_soff_alt	slave	Signal	-
trig_soff	slave	Signal	-
warn_event_ack	master	Signal	-
warn_event	slave	Signal	-

## Parameters for CCSM\_F1

### CFMM.CFM.diagnostics

Diagnostics.

Type: `uint8_t`

Default value: 2

### **CFMM.CMM.diagnostics**

Diagnostics.

Type: `uint8_t`

Default value: 2

### **CFMM.CMM.throttler\_num**

Number of input/output core clocks, and corresponding throttlers.

Type: `uint8_t`

Default value: 1

### **CFMM.CMM.warn\_num**

Number of warning events.

Type: `uint8_t`

Default value: 1

### **CFMM.diagnostics**

Diagnostics.

Type: `uint8_t`

Default value: 2

### **DVFSM.diagnostics**

Diagnostics.

Type: `uint8_t`

Default value: 2

### **DVFSM.num\_dvfsm\_ctrl\_out**

Number of bits in DVFSM\_CTRL\_OUT output.

Type: `uint8_t`

Default value: 1

### **DVFSM.num\_dvfsm\_data\_in**

Number of DVFSM\_DATA\_IN inputs.



Type: `uint8_t`

Default value: 1

**DVFSM.num\_dvfsm\_dynamic\_out**

Number of DVFSM\_DYNAMIC\_OUT outputs.

Type: `uint8_t`

Default value: 7

**DVFSM.num\_dvfsm\_event\_in**

Number of bits in DVFSM\_EVENT\_IN inputs.

Type: `uint8_t`

Default value: 3

**DVFSM.num\_dvfsm\_pse\_instr**

Maximum number of PSE instructions supported: 64 or 128.

Type: `uint8_t`

Default value: 128

**DVFSM.num\_dvfsm\_static\_out**

Number of DVFSM\_STATIC\_OUT outputs.

Type: `uint8_t`

Default value: 3

**DVFSM.num\_pll\_dynamic\_out**

Number of PLL Dynamic Setting outputs.

Type: `uint8_t`

Default value: 3

**DVFSM.num\_pll\_static\_out**

Number of PLL Static Setting outputs.

Type: `uint8_t`

Default value: 5

**DVFSM.num\_pll\_status\_in**

Number of PLL Status inputs.

Type: `uint8_t`

Default value: 1

### **`ccsm_clk_dd_div`**

Division parameter for CLK\_DD relative to CLK\_NOMM: 0 - no division, 1 - divide by 2 (currently supports only 0).

Type: `uint8_t`

Default value: 0

### **`diagnostics`**

Diagnostics.

Type: `uint8_t`

Default value: 2

### **`num_ccsm_ctrl_out`**

Number of bits in CCSM\_CTRL\_OUT output.

Type: `uint8_t`

Default value: 1

### **`num_ccsm_data_in`**

Number of CCSM\_DATA\_IN inputs.

Type: `uint8_t`

Default value: 1

### **`num_ccsm_dynamic_out`**

Number of CCSM\_DYNAMIC\_OUT outputs.

Type: `uint8_t`

Default value: 6

### **`num_ccsm_event_in`**

Number of bits in CCSM\_EVENT\_IN inputs.

Type: `uint8_t`

Default value: 2

**num\_ccsm\_ext\_warn**

Number of Warning Events for CCSM CMM.

Type: uint8\_t

Default value: 1

**num\_ccsm\_static\_out**

Number of CCSM\_STATIC\_OUT outputs.

Type: uint8\_t

Default value: 3

**num\_cffm**

Number of CFMMs per CCSM.

Type: uint8\_t

Default value: 1

**num\_clk\_throttlers**

Number of modulators/output clocks per CMM/CFMM.

Type: uint8\_t

Default value: 1

**num\_dvfsm\_pse\_instr**

Maximum number of PSE instructions supported: 64 or 128.

Type: uint8\_t

Default value: 128

**num\_pll\_dynamic\_out**

Number of PLL Dynamic Setting outputs.

Type: uint8\_t

Default value: 3

**num\_pll\_static\_out**

Number of PLL Static Setting outputs.

Type: uint8\_t

Default value: 5

**num\_pll\_status\_in**

Number of PLL Status inputs.

Type: uint8\_t

Default value: 1

## 3.133 CFM

Defined in `LISA/CFM.lisa`.

### About CFM

Clock Frequency based Mitigator (CFM).

### Iris and MTI instances for CFM

This model has the following Iris instances:

Name	Instance type
CFM	CFM
CFM.CfmMaskOrGate	WideOrGate
CFM.TrigOrGate	WideOrGate
CFM.clksel_mux	Value_Multiplexer
CFM.clock_mux	Clock_Multiplexer
CFM.fw_clksel_mux	Value_Multiplexer
CFM.trig_droop_mux	Signal_Multiplexer
CFM.trig_soff_mux	Signal_Multiplexer

### Ports for CFM

Port	Direction	Protocol	Description
cfm_clksel_ack	master	Signal	-
cfm_clksel_cur	master	ValueState	-
cfm_clksel_override_req	slave	ValueState	-
cfm_clksel_override_val	slave	ValueState	-
cfm_mask	slave	ValueState	-
cfm_override_applied	master	Signal	-
clk_dd_in	slave	ClockSignal	-
clk_fallback_in	slave	ClockSignal	-
clk_nominal_in	slave	ClockSignal	-
clkout_fm	master	ClockSignal	-
irq_err	master	Signal	-
PORESETn	slave	Signal	-
reg_pvbus_s	slave	PVBus	-

Port	Direction	Protocol	Description
RESETn	slave	Signal	-
sysclk_in	slave	ClockSignal	-
trig_droop_alt	slave	Signal	-
trig_droop_main	slave	Signal	-
trig_overshoot	slave	Signal	-
trig_soff_alt	slave	Signal	-
trig_soff_main	slave	Signal	-
warn_exit_fb_ack	slave	Signal	-
warn_exit_fb	master	Signal	-
warn_exit_stop_ack	slave	Signal	-
warn_exit_stop	master	Signal	-

## Parameters for CFM

### diagnostics

Diagnostics.

Type: uint8\_t

Default value: 2

## 3.134 CFMM

Defined in `LISA/CFMM.lisa`.

### About CFMM

Clock Frequency-based and Modulation-based Mitigator.

### Iris and MTI instances for CFMM

This model has the following Iris instances:

Name	Instance type
CFMM	CFMM
CFMM.CFM	CFM
CFMM.CFM.CfmMaskOrGate	WideOrGate
CFMM.CFM.TrigOrGate	WideOrGate
CFMM.CFM.clksel_mux	Value_Multiplexer
CFMM.CFM.clock_mux	Clock_Multiplexer
CFMM.CFM.fw_clksel_mux	Value_Multiplexer
CFMM.CFM.trig_droop_mux	Signal_Multiplexer
CFMM.CFM.trig_soff_mux	Signal_Multiplexer

Name	Instance type
CFMM.CMM	CMM
CFMM.CMM.throttler0	Throttler

## Ports for CFMM

Port	Direction	Protocol	Description
cfm_clkssel_cur	master	ValueState	-
cfm_clkssel_override_ack	master	Signal	-
cfm_clkssel_override_req	slave	ValueState	-
cfm_clkssel_override_val	slave	ValueState	-
cfm_irq	master	Signal	-
cfm_mask_ack	master	Signal	-
cfm_mask_req	slave	Signal	-
clk_dd_in	slave	ClockSignal	-
clk_fallback_in	slave	ClockSignal	-
clk_mm_out	master	ClockSignal	-
clk_nominal_in	slave	ClockSignal	-
clkout_fm	master	ClockSignal	-
PORESETn	slave	Signal	-
reg_pvbus_s	slave	PVBus	-
RESETn	slave	Signal	-
spare_in	slave	Signal	-
spare_out	master	Signal	-
sysclk_in	slave	ClockSignal	-
sysclk_qactive	master	Signal	-
trig_droop_alt	slave	Signal	-
trig_droop_main	slave	Signal	-
trig_overshoot	slave	Signal	-
trig_soff_alt	slave	Signal	-
trig_soff_main	slave	Signal	-
warn_ack	master	Signal	-
warn_event	slave	Signal	-

## Parameters for CFMM

### CFM.diagnostics

Diagnostics.

Type: uint8\_t

Default value: 2

**CMM.diagnostics**

Diagnostics.

Type: `uint8_t`

Default value: 2

**CMM.throttler\_num**

Number of input/output core clocks, and corresponding throttlers.

Type: `uint8_t`

Default value: 1

**CMM.warn\_num**

Number of warning events.

Type: `uint8_t`

Default value: 1

**diagnostics**

Diagnostics.

Type: `uint8_t`

Default value: 2

## 3.135 CHBCR

Defined in `LISA/CHBCR.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
0	Alpha support

For an explanation of the quality levels, see [Quality level definitions](#).

### About CHBCR

CXL Host Bridge Component Registers.

### Iris and MTI instances for CHBCR

This model has the following Iris instances:

Name	Instance type
CHBCR	CHBCR

### Ports for CHBCR

Port	Direction	Protocol	Description
apb_bus_s	slave	PVBus	-
chbcr_map_interrupt_out	master	Signal	-

### Parameters for CHBCR

#### **diagnostics**

Diagnostics.

Type: `uint8_t`

Default value: 2

#### **num\_decoders**

Number of Decoders.

Type: `uint8_t`

Default value: 16

#### **num\_ports**

Number of target ports.

Type: `uint8_t`

Default value: 4

## 3.136 CI700

Defined in `LISA/CI700.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r2p0	Full support
r1p0	Full support
r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).



## About CI700

- Major IP revisions (rX) are modeled and are controlled by the `version` parameter in the topology file. If topology doesn't contain this parameter, then model param 'revision' is used.
- Minor IP revisions (pY) are not modeled. There are no functional differences in the model between pY revisions, with the exception of the `periph_id2` register.
- The topology YAML 'version' field overrides the model parameter 'revision' when selecting the CMN revision.
- To configure the model, you must have installed Arm Socrates. The `mesh_config_file` parameter defines the mesh placement of the CHI nodes. Set it to the name of the yaml configuration file emitted by the Socrates export process. Fast Models requires the configuration file to pass Design Rule Check (DRC) in Socrates and does not support manually editing the configuration file. You must use version r1p6-00rel0 or later of Socrates to generate the file. For more information about Socrates, see [Arm Socrates](#) on Arm Developer. To download it, contact Arm Technical Support.
- Interconnect models are based on the TRM description and do not typically model RTL defects.

The model supports the following features:

- `rnf`, `rni/rnd`, `hni`, and `snf/sbsx` interface ports. The mapping between the port number and `NodeId` is based on the `NodeId` index. For example, `RNF2` controls `pvbuss_s_rnf[2]`. Its index is specified in the `node_info` register as `logical_id`. Similar behavior can be expected for `HN-I`.

If both `RN-D` and `RN-I` nodes are present, then all starting `rni` ports are mapped to `RN-D` nodes and then the `RN-I` nodes. For example, with two `RN-D` nodes, one `RN-I` node, and given each `RN-I` or `RN-D` node controls three interface ports, `pvbuss_s_rni[0-2]` maps to `RND0`, `pvbuss_s_rni[3-5]` maps to `RND1` and `pvbuss_s_rni[6-8]` maps to `RNI0`.

Similarly, `SN-F` and `SBSX` nodes are mapped to `pvbuss_m_snf[]` ports where starting ports are mapped to `SN-F` and then `SBSX` nodes.

- Mapping `HN*_SLC_SIZE_PARAM` values to cache sizes:

**-8**

OKB where `HN*_SLC_NUM_WAYS_PARAM=16`



**Note**

This value is not supported by the model.

---

**-2**

128KB where `HN*_SLC_NUM_WAYS_PARAM=16`

**-1**

256KB where `HN*_SLC_NUM_WAYS_PARAM=16`

**0**

512KB where `HN*_SLC_NUM_WAYS_PARAM=16`

**1**1MB where `HN*_SLC_NUM_WAYS_PARAM=16`**2**2MB where `HN*_SLC_NUM_WAYS_PARAM=16`**3**3MB where `HN*_SLC_NUM_WAYS_PARAM=12`**3**4MB where `HN*_SLC_NUM_WAYS_PARAM=16`

- The Address Based Flush (ABF) feature is available. After it has started, the ABF completes instantly. The ABF implementation has the following limitations:
  - In the Fast Model, any HN-F in the interconnect can perform an ABF on any address range even if that range is not part of the System Cache Group (SCG) that the HN-F belongs to.
  - Clearing the OCM with an ABF is not supported. A trace has been added to highlight this.
  - When `HN*_ABF_PR.abf_mode` is Reserved and `HN*_ABF_PR.abf_enable` is set, an error is printed and the ABF status register is set to complete successfully, but no operation takes place.
  - Because an ABF completes instantly, the `SF_PM_TRANSITION_ABORT` state cannot be reached. Contact Arm Technical Support with questions about support for this error condition.
  - The optional interrupt `INTREQPPU` is not supported.

## Limitations

- Out of scope:
  - PMU counters are not supported. Counter registers are implemented as **RAZ**.
  - QoS is not supported and all related registers are **RAZ/WI**.
  - Error injection and error generation are not supported. All error registers are **RAZ/WI**.
  - Power, clock, and interrupt signals are not supported.

The P-channel (power) signals `PREQ_LOGIC`, `PSTATE_LOGIC`, `PACCEPT_LOGIC`, `PDENY_LOGIC`, and `PACTIVE_LOGIC`, are not implemented. The model behaves as if `PSTATE_LOGIC` is 5'b00000. So the initial power state of the HN-F nodes is NOSFSLC/OFF. This is reflected in the reset values of `por_hnf_ppu_pwpr`. Software must write the `*_ppu_pwpr` registers to change the power state. Both HAM and FAM power states enable all of the SLC ways.

- Protocol credits and flit buffers are not supported.
- Snoop filtering is not supported.
- Prefetch Target operations are not supported.
- Non-XY routing behavior is not supported.
- The MTE feature has been minimally tested and the functionality cannot be guaranteed. MTE error injection and detection are not supported.

- An access to a reserved or nonexistent register does not abort. It returns `pv::Tx_Data::TX_OK` and is **RAZ/WI**.
- Topologies with three I/O masters connected through an RN-I bridge device have been tested. Configurations where multiple I/O masters share a single ACE-lite port have not been tested.
- RNSAMs external to the mesh network are not functionally supported.



Model parameter `force_rnsam_internal` is true by default. If this parameter is set to false and the topology has an RNFx (non-ESAM), the external bit (bit[31]) of the RNSAM children pointer is enabled. This bit is the only functionality governed by this parameter.

- By default, HN-F hashing uses the address [MAX:12] instead of the actual address [MAX:6], due to the DMI mechanism in the model. Enable the parameter `enable_rnsam_to_hnf_wider_hash` to make hashing logic use the actual address [MAX:6], but this might reduce the simulation speed of the model.
- OKB SLC data RAM and tag RAM are not supported.
- Source-based SLC cache partitioning is not supported.
- Way-based SLC cache partitioning is not supported.
- SLC/LCC partitioning dynamic allocation is not supported.
- Remote chip addresses incorrectly allocate a line in the SLC when `cache_state_modelled=true`. This can cause unexpected cache behavior that does not match the RTL, exhausting the SLC earlier or extra evictions.
- The fields in the `mxp_device_port_connect_info` register corresponding to the number of device credit slices and the number of CAL credit slices are not supported.
- SN-F hashing in HNSAM is not supported. HNSAM SN-F hashing is a single memory range mapped to multiple SN-Fs. Example HNFSAM\_Mem\_Map:

```
0x00000001000 - 0x00000002000 SNF0 SNF1
```

The model supports mapping HN-F to a single SN-F. Example HNFSAM\_Mem\_Map:

```
0x00000001000 - 0x00000002000 SNF0
0x00000002000 - 0x00000003000 SNF1
```

- HN-Fs with different SLC sizes in the same configuration are not supported.
- Transactions to addresses unused by Device registers in the Configuration Register Space are always routed to the Configuration Space even when the SAM is configured to use those addresses.
- GIC communication over A4S ports is not supported.
- MPAM features are not supported in the model, and the configuration-dependent register reset values do not match the RTL. Some of the registers (`por_hnf_mpam_iidr` to `por_hnf_mpam_mbwumon_iidr`) shared between security modes are not present in MPAM\_S.

- The MPAM reset values described in `por_hnf_mpam_ns_por_hnf_mpam_idr[31:24]` default to the not-supported state. The model parameter `hnf_mpam_idr_override` can be used to override this but does not implement any of the functionality.
- MPAM\_S `secure_register_groups_override` is not supported.
- MTU `secure_register_groups_override` is not supported.
- No specific Debug Trace (DT) node functionality is supported. See [Plug-ins for Fast Models](#) for information about using plug-ins for trace.
- For Power State Transition related flush/disable of the SLC due to HNF\_PPU\_PWPR register, only the Operational Mode field (bits[7:4]) is considered. The Power Policy field (bits[3:0]) is ignored.
- Setting HAM mode for the Operational Mode field (bits[7:4]) of the HNF\_PPU\_PWPR is equivalent to FAM mode. The SLCH2 is neither flushed nor disabled according to the Power Policy field (bits[3:0]).
- Only hashing granularities of 4096B and above are supported.
- The OCM does not keep track of the total memory used across the S/NS security world and does not perform any eviction either. It is up to the software to make sure it does not oversubscribe the OCM.
- All RNs are assumed to have the same HN mapping. Programming RNSAMs differently between RNs is not supported.
- RNSAM cannot be programmed with the same HN-F target in a hashed and non-hashed memory region.
- `RNSAM_STATUS.default_target_type` and `RNSAM_STATUS.default_nodeid` are used as the default target only when `RNSAM_STATUS.use_default_node=1`. Otherwise, when a txn is sent into the model that does not belong to any of the address regions programmed in the RNSAM, it is routed to the HN-D irrespective of what is programmed in the `RNSAM_STATUS.default_target_type` and `RNSAM_STATUS.default_nodeid` fields.
- Address bit masking is not supported in RNSAM or HN-F SAM.
- For an SCG, the model does not consider the secondary region if the primary region is not valid.
- HN-D is only permitted on device port P2 in a single-MXP configuration.
- No support for RAS.
- Only the last RN-F in the mesh, which is the one with the highest logical id, can be controlled by `mxp_p[0-5]_syscoreq_ctl` registers. Also, incorrectly, it can be controlled from any XP.
- HN-F SAM limitations:
  - Address masking in default hash regions in HN-F SAM is not supported.
  - Hashing in default hash regions in HN-F SAM is not supported. However, trace `SNF_HASHING_Target_SNF` displays the target SNF as if 3SN/6SN hashing were enabled.



The transaction is always routed to the SNO nodeid programmed in the SAM\_CONTROL register.

- Re-programming regions in HN-F SAM is not tested.
- POR\_MTU\_TAG\_ADDR\_CTL.memory\_map\_mode=3'b000 (Pass-through) is the only supported behavior. Other values for this field are not supported.
- For RN-D nodes, when software writes SYSCOREQ, DVM propagation gets enabled but SYSCOACK is not set.

### About the debug\_force\_snoop parameter

The interconnect model normally starts up with snooping disabled.

The parameters `rnf_sci_enable` and `rni_sci_enable` determine which connections are managed by the System Coherency Interface. Connections that are managed by the System Coherency Interface look after themselves for entering and leaving the coherency domain and `debug_force_snoop` has no effect on those ports.

However, for connections that are not managed by the System Coherency Interface, `debug_force_snoop` enables the system upstream of a port to be snooped if the upstream is not in reset and if `acchannelen_rnf` allows it.

Therefore, software does not need to know how to turn snoops in the interconnect on and off. The reset state of the upstream `rnf` ports is determined by the `rnf_upstream_reset_state[]` signals and must be connected for any port that could go into reset and you want managed by `debug_force_snoop`.

If the connection is managed by SCI or can never go into reset then you need not connect this and the interconnect assumes that the upstream system is not in reset and will always enable the snoops if `ACCHANNELENSx` allows it.

If you fail to connect `rnf_upstream_reset_state[]` correctly, then the upstream system might receive snoop messages while in reset and it will complain that it received a snoop request while it was in reset.

If software overrides the enables then the next change to `rnf_upstream_reset_state[]` or the interconnect reset will overwrite the software values.

Using this operation along with software controlling the enables could confuse the software. This option does not remove the responsibility of the upstream system to clean any shared dirty data from its caches before going into reset.



This parameter is for debug purposes only. Do not use it instead of correctly configuring the model.

## Iris and MTI instances for CI700

This model has the following Iris instances:

Name	Instance type
CI700	CI700
CI700.bus_slave_ocm_NS	PVBusSlave
CI700.bus_slave_ocm_S	PVBusSlave
CI700.ci700_tag_cache	CMN_TAG_CACHE
CI700.ci700_tag_cache.metadata_controllerZ (where Z = 0-127)	PVMetadataController
CI700.ci700_tag_cache.metadata_controllerZ.MetaDataMapper (where Z = 0-127)	PVBusMapper
CI700.ci700_tag_cache.metadata_controllerZ.mdc_downstream_port[0].pvbusmaster (where Z = 0-127)	PVBusMaster
CI700.ci700_tag_cache.remapperZ (where Z = 0-127)	PVBusMapper
CI700.cmn600_cache	PVCache
CI700.cmn600_cache.downstream[Z].pvbusmaster (where Z = 0-7)	PVBusMaster
CI700.cmn600_cache.upstream[Z] (where Z = 0-3)	PVBusSlave
CI700.hnf_exclusive_monitor	PVBusExclusiveMonitor
CI700.hnf_exclusive_monitor.bus_mapper	PVBusMapper
CI700.hni_exclusive_monitor0	PVBusExclusiveMonitor
CI700.hni_exclusive_monitor0.bus_mapper	PVBusMapper
CI700.ocm_decoder	PVBusMapper
CI700.ocm_exclusive_monitor	PVBusExclusiveMonitor
CI700.ocm_exclusive_monitor.bus_mapper	PVBusMapper
CI700.snf_mapper	PVBusMapper

This model has the following MTI trace components:

Name	Component type
CI700	CI700
CI700.bus_slave_ocm_NS	PVBusSlave
CI700.bus_slave_ocm_S	PVBusSlave
CI700.ci700_tag_cache	CMNTAGCACHECADI
CI700.ci700_tag_cache.metadata_controllerZ (where Z = 0-127)	MetaDataController
CI700.ci700_tag_cache.metadata_controllerZ.MetaDataMapper (where Z = 0-127)	PVBusMapper
CI700.ci700_tag_cache.metadata_controllerZ.mdc_downstream_port[0].pvbusmaster (where Z = 0-127)	PVBusMaster
CI700.ci700_tag_cache.remapperZ (where Z = 0-127)	PVBusMapper
CI700.cmn600_cache	CMN600Cache

Name	Component type
CI700.cmn600_cache.downstream[Z].pvbusmaster (where Z = 0-7)	PVBusMaster
CI700.cmn600_cache.upstream[Z] (where Z = 0-3)	PVBusSlave
CI700.hnf_exclusive_monitor	PVBusExclusiveMonitor
CI700.hnf_exclusive_monitor.bus_mapper	PVBusMapper
CI700.hni_exclusive_monitor0	PVBusExclusiveMonitor
CI700.hni_exclusive_monitor0.bus_mapper	PVBusMapper
CI700.ocm_decoder	PVBusMapper
CI700.ocm_exclusive_monitor	PVBusExclusiveMonitor
CI700.ocm_exclusive_monitor.bus_mapper	PVBusMapper
CI700.snf_mapper	PVBusMapper

## Ports for CI700

Port	Direction	Protocol	Description
event_clusters	peer	Signal	CPU event communication signal from the clusters.
pvbus_m_hni	master	PVBus	HNI downstream ports.
pvbus_m_snf	master	PVBus	SNF downstream port.
pvbus_s_apb	slave	PVBus	APB interface port.
pvbus_s_rnf	slave	PVBus	RNF upstream ports.
pvbus_s_rni	slave	PVBus	RNI upstream ports.
reset_in	slave	Signal	Reset signal.
rnf_sci_s	slave	SystemCoherencyInterface	System Coherency port to move the RN-F in and out of the coherency domain.
rnf_upstream_reset_in	slave	Signal	Used by the interconnect to determine the reset state of the RNF manager to check an aspect of reset sequencing. If debug_force_snoop is used, this port must be connected.
rni_sci_s	slave	SystemCoherencyInterface	System Coherency port to move the RN-D in and out of the coherency domain.
rni_upstream_reset_in	slave	Signal	Used by the interconnect to determine the reset state of the RNI manager to check an aspect of reset sequencing. If debug_force_snoop is used, this port must be connected.

## Parameters for CI700

### acchannelen\_rnf



DEPRECATED: Will be removed after FM 11.18. Use `rnf_sci_enable` instead.

For each rnf port, indicates if the port is populated with a snoop responding device or not.

The input value is a string, for example `0xffff` or `ffff`.

Type: `string`

Default value: `"0"`

### **acchannelen\_rni**

---



Note

DEPRECATED: Will be removed after FM 11.18. Use `rni_sci_enable` instead.

---

For each `rni` port, indicates if the port is populated with a dvm responding device or not.

The input value is a string, for example `0xffff` or `ffff`.

Type: `string`

Default value: `"0"`

### **bypass\_tag\_cache**

If true, CI700 will bypass the tag cache component which provides the MTE support.

Type: `bool`

Default value: `false`

### **cache\_state\_modelled**

Model the cache state.

Type: `bool`

Default value: `false`

### **debug\_force\_snoop**

The CI700 interconnect will normally start with snooping disabled.

The parameter `rnf_sci_enable` and `rni_sci_enable` determines which connections are managed by the System Coherency Interface.

For connections that are managed by the System Coherency Interface, then they will look after themselves for entering and leaving the coherency domain and this parameter has no effect for those ports.

However, for connections that are *not* managed by the System Coherency Interface, this parameter enables the upstream system of a port to be snooped if the upstream is not in reset and if `acchannelen_rnf` allows it.





This parameter is for debug purpose only

Do not use this parameter instead of correctly configuring CMN.

---

Type: `bool`

Default value: `false`

### **`enable_logger`**

Enable PVBUSLoggers for the downstream ports in the CMN model.

Type: `bool`

Default value: `false`

### **`enable_rnsam_to_hnf_wider_hash`**

Enable support of wider hash for the RNSAM to HNF communication.

If this variable enabled, then `bits[47:6]` from PA used in hashing function.

By default it is `[47:12]`.

Type: `bool`

Default value: `false`

### **`force_rnsam_internal`**

Force all RNSAMs to be internal independently of the mesh topology.

Type: `bool`

Default value: `true`

### **`hnf_mpam_idr_override`**

Set to override `hnf_mpam_idr[31:24]` value. Bit[28] is Reserved and is ignored.

Type: `uint64_t`

Default value: `0`

### **`mesh_config_file`**

Name of a file containing mesh placement of CI700 components.

Type: `string`

Default value: `""`

**periphbase**

Value for PERIPHBASE. Bits [27:0] are treated 0.

Type: `uint64_t`

Default value: `0x20000000`

**print\_cmn\_ccix\_config**

Print information about the CCIX configuration.

Type: `bool`

Default value: `false`

**print\_cmn\_config**

Print the mesh topology and children pointers acquired from the YML file.

Type: `bool`

Default value: `false`

**revision**

Component revision.

Currently supports r2p0, r1p0, r0p0.

Type: `string`

Default value: `"r0p0"`

**rnf\_sci\_enable**

For each rnf port, indicates if the port is managed by System Coherency Interface for coherency domain entry/exit

**1**

Managed by SCI

**0**

Managed by Software

The input value is a string, for example `0xffff` or `ffff`.

Type: `string`

Default value: `"0x0"`

**rni\_sci\_enable**

For each rni port, indicates if the port is managed by System Coherency Interface for coherency domain entry/exit

**1**

Managed by SCI

**0**

Managed by Software

The input value is a string, for example `0xffff` or `ffff`.

Type: `string`

Default value: `"0x0"`

**show\_banner**

Show component banner:

**0**

supress entire banner

**1**

suppress config file

**2+**

show full banner.

Type: `uint64_t`

Default value: 2

**skip\_cmn\_config\_check**

Skip any topology configuration checks. The maximum number of devices per type not verified.

Type: `bool`

Default value: `false`

**use\_yaml\_periphbase**

Use yaml param `CFGM_PERIPHBASE_PARAM` to specify periphbase address.

If false, model parameter `periphbase` will be used.

Type: `bool`

Default value: `false`

**yaml\_has\_node\_addresses**

Does the top-level YML file describe node-addresses ?.

Type: `bool`

Default value: `false`

## 3.137 CMM

Defined in `LISA/CMM.lisa`.

### About CMM

Clock Modulation based Mitigator (CMM).

### Iris and MTI instances for CMM

This model has the following Iris instances:

Name	Instance type
CMM	CMM
CMM.throttler0	Throttler

### Ports for CMM

Port	Direction	Protocol	Description
clkin_mm	slave	ClockSignal	-
clkout_mm	master	ClockSignal	-
reg_pvbus_s	slave	PVBus	-
RESETn	slave	Signal	-
sysclk_in	slave	ClockSignal	-
warn_ack	master	Signal	-
warn_event	slave	Signal	-

### Parameters for CMM

#### **diagnostics**

Diagnostics.

Type: `uint8_t`

Default value: 2

#### **throttler\_num**

Number of input/output core clocks, and corresponding throttlers.

Type: `uint8_t`

Default value: 1

**warn\_num**

Number of warning events.

Type: `uint8_t`

Default value: 1

## 3.138 CMN600

Defined in `LISA/CMN600.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r1p1	Full support
r3p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### About CMN600

- Major IP revisions (rX) are modeled and are controlled by the `revision` parameter.
- Minor IP revisions (pY) are not modeled. There are no functional differences in the model between pY revisions, with the exception of the `periph_id2` register.
- The topology YAML 'version' field overrides the model parameter 'revision' when selecting the CMN revision.
- To configure the model, you must have installed Arm Socrates. The `mesh_config_file` parameter defines the mesh placement of the CHI nodes. Set it to the name of the yaml configuration file emitted by the Socrates export process. Fast Models requires the configuration file to pass Design Rule Check (DRC) in Socrates and does not support manually editing the configuration file. You must use version r1p4-01rel0 or later of Socrates to generate the file. For more information about Socrates, see [Arm Socrates](#) on Arm Developer. To download it, contact Arm Technical Support.
- Interconnect models are based on the TRM description and do not typically model RTL defects.

The model supports the following features:

- `rnf`, `rni/rnd`, `hni`, and `snf/sbsx` interface ports. The mapping between the port number and `nodeId` is based on the `nodeId` index. For example, RNF2 controls `pvbuss_s_rnf[2]`. Its index is specified in the `node_info` register as `logical_id`. Similar behavior can be expected for HN-I.

If both RN-D and RN-I nodes are present, then all starting `rni` ports are mapped to RN-D nodes and then the RN-I nodes. For example, for CMN600 with two RN-D nodes, one RN-I node, and given each RN-I or RN-D node controls three interface ports, `pvbuss_s_rni[0-2]` maps to RND0, `pvbuss_s_rni[3-5]` maps to RND1 and `pvbuss_s_rni[6-8]` maps to RNI0.

Similarly, SN-F and SBSX nodes are mapped to `pvtbus_m_snf[]` ports where starting ports are mapped to SN-F and then SBSX nodes.

- Mapping HN\*\_SLC\_SIZE\_PARAM values to cache sizes:

-8

OKB where HN\*\_SLC\_NUM\_WAYS\_PARAM=16



Note

This value is not supported by the model.

---

-2

128KB where HN\*\_SLC\_NUM\_WAYS\_PARAM=16

-1

256KB where HN\*\_SLC\_NUM\_WAYS\_PARAM=16

0

512KB where HN\*\_SLC\_NUM\_WAYS\_PARAM=16

1

1MB where HN\*\_SLC\_NUM\_WAYS\_PARAM=16

2

2MB where HN\*\_SLC\_NUM\_WAYS\_PARAM=16

3

3MB where HN\*\_SLC\_NUM\_WAYS\_PARAM=12

3

4MB where HN\*\_SLC\_NUM\_WAYS\_PARAM=16

- The Address Based Flush (ABF) feature is available. After it has started, the ABF completes instantly. The ABF implementation has the following limitations:
  - In the Fast Model, any HN-F in the interconnect can perform an ABF on any address range even if that range is not part of the System Cache Group (SCG) that the HN-F belongs to.
  - Clearing the OCM with an ABF is not supported. A trace has been added to highlight this.
  - When `HN*_ABF_PR.abf_mode` is Reserved and `HN*_ABF_PR.abf_enable` is set, an error is printed and the ABF status register is set to complete successfully, but no operation takes place.
  - Because an ABF completes instantly, the `SF_PM_TRANSITION_ABORT` state cannot be reached. Contact Arm Technical Support with questions about support for this error condition.
  - The optional interrupt `INTREQFPU` is not supported.

## Limitations

- Out of scope:
  - PMU counters are not supported. Counter registers are implemented as **RAZ**.

- QoS is not supported and all related registers are **RAZ/WI**.
- Error injection and error generation are not supported. All error registers are **RAZ/WI**.
- Power, clock, and interrupt signals are not supported.

The P-channel (power) signals PREQ\_LOGIC, PSTATE\_LOGIC, PACCEPT\_LOGIC, PDENY\_LOGIC, and PACTIVE\_LOGIC, are not implemented. The model behaves as if PSTATE\_LOGIC is 5'b00000. So the initial power state of the HN-F nodes is NOSFSLC/OFF. This is reflected in the reset values of `cmn_hns_ppu_pwpr`, which was named `por_hnf_ppu_pwpr` in earlier IP revisions. Software must write the `*_ppu_pwpr` registers to change the power state. Both HAM and FAM power states enable all of the SLC ways.

- Protocol credits and flit buffers are not supported.
- Snoop filtering is not supported.
- Prefetch Target operations are not supported.
- The MTE feature has been minimally tested and the functionality cannot be guaranteed. MTE error injection and detection are not supported.
- An access to a reserved or nonexistent register does not abort. It returns `pv::Tx_Data::TX_OK` and is **RAZ/WI**.
- Topologies with three I/O masters connected through an RN-I bridge device have been tested. Configurations where multiple I/O masters share a single ACE-lite port have not been tested.
- RNSAMs external to the mesh network are not functionally supported.



Model parameter `force_rnsam_internal` is true by default. If this parameter is set to false and the topology has an RNFx (non-ESAM), the external bit (bit[31]) of the RNSAM children pointer is enabled. This bit is the only functionality governed by this parameter.

- By default, HN-F hashing uses the address [MAX:12] instead of the actual address [MAX:6], due to the DMI mechanism in the model. Enable the parameter `enable_rnsam_to_hnf_wider_hash` to make hashing logic use the actual address [MAX:6], but this might reduce the simulation speed of the model.
- OKB SLC data RAM and tag RAM are not supported.
- Source-based SLC cache partitioning is not supported.
- Way-based SLC cache partitioning is not supported.
- SLC/LCC partitioning dynamic allocation is not supported.
- Remote chip addresses incorrectly allocate a line in the SLC when `cache_state_modelled=true`. This can cause unexpected cache behavior that does not match the RTL, exhausting the SLC earlier or extra evictions.
- The fields in the `mvp_device_port_connect_info` register corresponding to the number of device credit slices and the number of CAL credit slices are not supported.

- SN-F hashing in HNSAM is not supported. HNSAM SN-F hashing is a single memory range mapped to multiple SN-Fs. Example HNFSAM\_Mem\_Map:

```
0x00000001000 - 0x00000002000 SNF0 SNF1
```

The model supports mapping HN-F to a single SN-F. Example HNFSAM\_Mem\_Map:

```
0x00000001000 - 0x00000002000 SNF0
0x00000002000 - 0x00000003000 SNF1
```

- HN-Fs with different SLC sizes in the same configuration are not supported.
- Transactions to addresses unused by Device registers in the Configuration Register Space are always routed to the Configuration Space even when the SAM is configured to use those addresses.
- GIC communication over A4S ports is not supported.
- CAL (Component Aggregation Layer) r2 and r3 features are supported. These features have limited testing.
- No specific Debug Trace (DT) node functionality is supported. See [Plug-ins for Fast Models](#) for information about using plug-ins for trace.
- For Power State Transition related flush/disable of the SLC due to HNF\_PPU\_PWPR register, only the Operational Mode field (bits[7:4]) is considered. The Power Policy field (bits[3:0]) is ignored.
- Setting HAM mode for the Operational Mode field (bits[7:4]) of the HNF\_PPU\_PWPR is equivalent to FAM mode. The SLCH2 is neither flushed nor disabled according to the Power Policy field (bits[3:0]).
- Only hashing granularities of 4096B and above are supported.
- All RNs are assumed to have the same HN mapping. Programming RNSAMs differently between RNs is not supported.
- RNSAM cannot be programmed with the same HN-F target in a hashed and non-hashed memory region.
- Address bit masking is not supported in RNSAM or HN-F SAM.
- The OCM does not keep track of the total memory used across the S/NS security world and does not perform any eviction either. It is up to the software to make sure it does not oversubscribe the OCM.
- In revision r3p0, for an HTG/SCG, the model does not consider the secondary region if the primary region is not valid.
- There is no support for RAS.
- The following limitations apply to System Cache Groups and Hash Target Groups:
  - A mix of local and remote HN-F targets is not supported.
  - CCG/CXRH target IDs in the HN target ID table are not supported (`sys_cache_grp_hn_nodeid_reg`).



- The `hashed_target_grp_hnp_nodeid_reg` which is used for CCG/HN-P target IDs is not supported.
- System/Hash Target Groups only support HN-Fs.
- AXID hashing across HN-P/CCGs is not supported.
- HN-F SAM:
  - Address masking in default hash regions in HN-F SAM is not supported.
  - Hashing in default hash regions in HN-F SAM is not supported. However, trace `SNF_HASHING_Target_SNF` displays the target SNF as if 3SN/6SN hashing were enabled.



The transaction is always routed to the SNO nodeid programmed in the SAM\_CONTROL register.

- 
- Re-programming regions in HN-F SAM is not tested.
  - Hashing across CCGs in HN-F SAM is not supported.
  - For RN-D nodes, when software writes SYSCOREQ, DVM propagation gets enabled but SYSCOACK is not set.

### About the `debug_force_snoop` parameter

The CMN interconnect normally starts up with snooping disabled.

The parameters `rnf_sci_enable` and `rni_sci_enable` determine which connections are managed by the System Coherency Interface. Connections that are managed by the System Coherency Interface look after themselves for entering and leaving the coherency domain and `debug_force_snoop` has no effect on those ports.

However, for connections that are not managed by the System Coherency Interface, `debug_force_snoop` enables the system upstream of a port to be snooped if the upstream is not in reset and if `acchannelen_rnf` allows it.

Therefore, software does not need to know how to turn snoops in the interconnect on and off. The reset state of the upstream rnf ports is determined by the `rnf_upstream_reset_state[]` signals and must be connected for any port that could go into reset and you want managed by `debug_force_snoop`.

If the connection is managed by SCI or can never go into reset then you need not connect this and the interconnect assumes that the upstream system is not in reset and will always enable the snoops if `ACCHANNELENSx` allows it.

If you fail to connect `rnf_upstream_reset_state[]` correctly, then the upstream system might receive snoop messages while in reset and it will complain that it received a snoop request while it was in reset.

If software overrides the enables then the next change to `rnf_upstream_reset_state[]` or the interconnect reset will overwrite the software values.

Using this operation along with software controlling the enables could confuse the software. This option does not remove the responsibility of the upstream system to clean any shared dirty data from its caches before going into reset.



This parameter is for debug purposes only. Do not use it instead of correctly configuring CMN.

## Iris and MTI instances for CMN600

This model has the following Iris instances:

Name	Instance type
CMN600	CMN600
CMN600.bus_slave_ocm_NS	PVBusSlave
CMN600.bus_slave_ocm_S	PVBusSlave
CMN600.cmn600_cache	CMN600Cache
CMN600.cmn600_cache.downstream[Z].pvbusmaster (where Z = 0-8)	PVBusMaster
CMN600.cmn600_cache.upstream[Z] (where Z = 0-17)	PVBusSlave
CMN600.hnf_exclusive_monitor	PVBusExclusiveMonitor
CMN600.hnf_exclusive_monitor.bus_mapper	PVBusMapper
CMN600.hni_exclusive_monitor0	PVBusExclusiveMonitor
CMN600.hni_exclusive_monitor0.bus_mapper	PVBusMapper
CMN600.ocm_decoder	PVBusMapper
CMN600.ocm_exclusive_monitor	PVBusExclusiveMonitor
CMN600.ocm_exclusive_monitor.bus_mapper	PVBusMapper
CMN600.snf_mapper	PVBusMapper

This model has the following MTI trace components:

Name	Component type
CMN600	CMN600
CMN600.bus_slave_ocm_NS	PVBusSlave
CMN600.bus_slave_ocm_S	PVBusSlave
CMN600.cmn600_cache	CMN600Cache
CMN600.cmn600_cache.downstream[Z].pvbusmaster (where Z = 0-8)	PVBusMaster
CMN600.cmn600_cache.upstream[Z] (where Z = 0-17)	PVBusSlave
CMN600.hnf_exclusive_monitor	PVBusExclusiveMonitor
CMN600.hnf_exclusive_monitor.bus_mapper	PVBusMapper
CMN600.hni_exclusive_monitor0	PVBusExclusiveMonitor

Name	Component type
CMN600.hni_exclusive_monitor0.bus_mapper	PVBusMapper
CMN600.ocm_decoder	PVBusMapper
CMN600.ocm_exclusive_monitor	PVBusExclusiveMonitor
CMN600.ocm_exclusive_monitor.bus_mapper	PVBusMapper
CMN600.snf_mapper	PVBusMapper

## Ports for CMN600

Port	Direction	Protocol	Description
event_clusters	peer	Signal	CPU event communication signal from the clusters.
event_downstream_link_signal	master	Signal	CPU event communication signal to the CMLHub. Event from the CMN towards the Hub NOTE that these are virtual “links” on underlying PCIe hardware bus.
event_upstream_link_signal	slave	Signal	Event from the Hub towards the CMN
pvbus_m_cml_cfg	master	PVBus	CML downstream hub configuration port
pvbus_m_cml	master	PVBus	CML downstream ports
pvbus_m_hni	master	PVBus	HNI downstream ports.
pvbus_m_snf	master	PVBus	SNF downstream port.
pvbus_s_cml	slave	PVBus	CML upstream ports
pvbus_s_rnf	slave	PVBus	RNF upstream ports.
pvbus_s_rni	slave	PVBus	RNI upstream ports.
reset_in	slave	Signal	Reset signal.
rnf_sci_s	slave	SystemCoherencyInterface	System Coherency port to move the RN-F in and out of the coherency domain.
rnf_upstream_reset_in	slave	Signal	Used by the interconnect to determine the reset state of the RNF manager to check an aspect of reset sequencing. If debug_force_snoop is used, this port must be connected.
rni_sci_s	slave	SystemCoherencyInterface	System Coherency port to move the RN-D in and out of the coherency domain.
rni_upstream_reset_in	slave	Signal	Used by the interconnect to determine the reset state of the RNI manager to check an aspect of reset sequencing. If debug_force_snoop is used, this port must be connected.

## Parameters for CMN600

### acchannelen\_rnf



DEPRECATED: Will be removed after FM 11.18. Use `rnf_sci_enable` instead.

For each rnf port, indicates if the port is populated with a snoop responding device or not.

The input value is a string, for example `0xffff` or `ffff`.

Type: `string`

Default value: `"0"`

### **`acchannelen_rni`**

---



Note

DEPRECATED: Will be removed after FM 11.18. Use `rni_sci_enable` instead.

---

For each `rni` port, indicates if the port is populated with a dvm responding device or not.

The input value is a string, for example `0xffff` or `ffff`.

Type: `string`

Default value: `"0"`

### **`cache_state_modelled`**

Model the cache state.

Type: `bool`

Default value: `false`

### **`debug_force_snoop`**

The CMN600 interconnect will normally start with snooping disabled.

The parameter `rnf_sci_enable` and `rni_sci_enable` determines which connections are managed by the System Coherency Interface.

For connections that are managed by the System Coherency Interface, then they will look after themselves for entering and leaving the coherency domain and this parameter has no effect for those ports.

However, for connections that are *not* managed by the System Coherency Interface, this parameter enables the upstream system of a port to be snooped if the upstream is not in reset and if `acchannelen_rnf` allows it.



Note

This parameter is for debug purpose only

Do not use this parameter instead of correctly configuring CMN.

---

Type: `bool`

Default value: `false`

### **`disable_CML_port`**

If true the model won't connect the CML port when there are nodes which support the CML feature (ie CXG nodes).

Type: `bool`

Default value: `false`

### **`enable_logger`**

Enable PVBUSLoggers for the downstream ports in the CMN model.

Type: `bool`

Default value: `false`

### **`enable_rnsam_to_hnf_wider_hash`**

Enable support of wider hash for the RNSAM to HNF communication.

If this variable enabled, then `bits[47:6]` from PA used in hashing function.

By default it is `[47:12]`.

Type: `bool`

Default value: `false`

### **`force_rnsam_internal`**

Force all RNSAMs to be internal independently of the mesh topology.

Type: `bool`

Default value: `true`

### **`mesh_config_file`**

Name of a file containing mesh placement of CMN600 components.

Type: `string`

Default value: `""`

### **`periphbase`**

Value for `PERIPHBASE`. Bits `[25:0]` are treated 0.

Type: `uint64_t`

Default value: `0x20000000`

**print\_cmn\_ccix\_config**

Print information about the CCIX configuration.

Type: `bool`

Default value: `false`

**print\_cmn\_config**

Print the mesh topology and children pointers acquired from the YML file.

Type: `bool`

Default value: `false`

**revision**

Component revision.

Currently supports r1p1, r3p0.

Type: `string`

Default value: `"r1p1"`

**rnf\_sci\_enable**

For each rnf port, indicates if the port is managed by System Coherency Interface for coherency domain entry/exit

**1**

Managed by SCI

**0**

Managed by Software

The input value is a string, for example `0xffff` or `ffff`.

Type: `string`

Default value: `"0x0"`

**rni\_sci\_enable**

For each rni port, indicates if the port is managed by System Coherency Interface for coherency domain entry/exit

**1**

Managed by SCI

**0**

Managed by Software

The input value is a string, for example `0xffff` or `ffff`.

Type: `string`

Default value: `"0x0"`

**show\_banner**

Show component banner:

**0**

supress entire banner

**1**

suppress config file

**2+**

show full banner.

Type: `uint64_t`

Default value: `2`

**skip\_cmn\_config\_check**

Skip any topology configuration checks. The maximum number of devices per type not verified.

Type: `bool`

Default value: `false`

**use\_yaml\_periphbase**

Use yaml param `CFGM_PERIPHBASE_PARAM` to specify periphbase address.

If false, model parameter `periphbase` will be used.

Type: `bool`

Default value: `false`

## 3.139 CMN600AE

Defined in `LISA/CMN600AE.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r1p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

## About CMN600AE

- Major IP revisions (rX) are modeled and are controlled by the `revision` parameter.
- Minor IP revisions (pY) are not modeled. There are no functional differences in the model between pY revisions, with the exception of the `periph_id2` register.
- The topology YAML 'version' field overrides the model parameter 'revision' when selecting the CMN revision.
- To configure the model, you must have installed Arm Socrates. The `mesh_config_file` parameter defines the mesh placement of the CHI nodes. Set it to the name of the yml configuration file emitted by the Socrates export process. Fast Models requires the configuration file to pass Design Rule Check (DRC) in Socrates and does not support manually editing the configuration file. You must use version r1p4-01rel0 or later of Socrates to generate the file. For more information about Socrates, see [Arm Socrates](#) on Arm Developer. To download it, contact Arm Technical Support.
- Interconnect models are based on the TRM description and do not typically model RTL defects.

The CMN600AE model supports the following features:

- `rnf`, `rni/rnd`, `hni`, and `snf/sbsx` interface ports. The mapping between the port number and `nodeId` is based on the `nodeId` index. For example, RNF2 controls `pvbuss_s_rnf[2]`. Its index is specified in the `node_info` register as `logical_id`. Similar behavior can be expected for HN-I.

If both RN-D and RN-I nodes are present, then all starting `rni` ports are mapped to RN-D nodes and then the RN-I nodes. For example, for CMN600AE with two RN-D nodes, one RN-I node, and given each RN-I or RN-D node controls three interface ports, `pvbuss_s_rni[0-2]` maps to RND0, `pvbuss_s_rni[3-5]` maps to RND1 and `pvbuss_s_rni[6-8]` maps to RNI0. Similarly, SN-F and SBSX nodes are mapped to `pvbuss_m_snf[]` ports where starting ports are mapped to SN-F and then SBSX nodes.

- Mapping HN\*\_SLC\_SIZE\_PARAM values to cache sizes:

-8

0KB when HN\*\_SLC\_NUM\_WAYS\_PARAM=16



Note

This value is not supported by the model.

-2

128KB when HN\*\_SLC\_NUM\_WAYS\_PARAM=16

-1

256KB when HN\*\_SLC\_NUM\_WAYS\_PARAM=16



**0**512KB when `HN*_SLC_NUM_WAYS_PARAM=16`**1**1MB when `HN*_SLC_NUM_WAYS_PARAM=16`**2**2MB when `HN*_SLC_NUM_WAYS_PARAM=16`**3**3MB when `HN*_SLC_NUM_WAYS_PARAM=12`**3**4MB when `HN*_SLC_NUM_WAYS_PARAM=16`

- Maximum number of nodes that have been verified are:
  - 7 RN-Fs
  - 2 RN-Ds
  - 3 RN-Is
  - 1 HN-I
  - 4 HN-Fs
  - 1 SN-F
- The Address Based Flush (ABF) feature is available. After it has started, the ABF completes instantly. The ABF implementation has the following limitations:
  - In the Fast Model, any HN-F in the interconnect can perform an ABF on any address range even if that range is not part of the System Cache Group (SCG) that the HN-F belongs to.
  - Clearing the OCM with an ABF is not supported. A trace has been added to highlight this.
  - When `HNF_ABF_PR.abf_mode` is Reserved and `HNF_ABF_PR.abf_enable` is set, an error is printed and the ABF status register is set to complete successfully, but no operation takes place.
  - Because an ABF completes instantly, the `SF_PM_TRANSITION_ABORT` state cannot be reached. Contact Arm Technical Support with questions about support for this error condition.
  - The optional interrupt `INTREQPPU` is not supported.
- This model implements the following AE-specific features:
  - Memory Protection Unit (MPU), with limitations listed in the Limitations section.
  - FuSa error logging and reporting using a Fault Management Unit (FMU) and Fault Detection and Control (FDC).
  - Dedicated APB interface into FMU for fault diagnostics and control.

## Limitations

- Out of scope:
  - PMU counters are not supported. Counter registers are implemented as **RAZ**.
  - QoS is not supported and all related registers are **RAZ/WI**.

- Error injection and error generation are not supported. All error registers are **RAZ/WI**.
- Power, clock, and interrupt signals are not supported.

The P-channel (power) signals `PREQ_LOGIC`, `PSTATE_LOGIC`, `PACCEPT_LOGIC`, `PDENY_LOGIC`, and `PACTIVE_LOGIC`, are not implemented. The model behaves as if `PSTATE_LOGIC` is `5'b00000`. So the initial power state of the HN-F nodes is `NOSFSLC/OFF`. This is reflected in the reset values of `cmn_hns_ppu_pwpr`, which was named `por_hnf_ppu_pwpr` in earlier IP revisions. Software must write the `*_ppu_pwpr` registers to change the power state. Both HAM and FAM power states enable all of the SLC ways.

- Protocol credits and flit buffers are not supported.
- Snoop filtering is not supported.
- Prefetch Target operations are not supported.
- The MTE feature has been minimally tested and the functionality cannot be guaranteed. MTE error injection and detection are not supported.
- An access to a reserved or nonexistent register does not abort. It returns `pv::Tx_Data::TX_OK` and is **RAZ/WI**.
- Topologies with three I/O masters connected through an RN-I bridge device have been tested. Configurations where multiple I/O masters share a single ACE-Lite port have not been tested.
- RNSAMs external to the mesh network are not functionally supported.



Note

Model parameter `force_rnsam_internal` is true by default. If this parameter is set to false and the topology has an RNFx (non-ESAM), the external bit (bit[31]) of the RNSAM children pointer is enabled. This bit is the only functionality governed by this parameter.

- By default, HN-F hashing uses the address `[MAX:12]` instead of the actual address `[MAX:6]`, due to the DMI mechanism in the model. Enable the parameter `enable_rnsam_to_hnf_wider_hash` to make hashing logic use the actual address `[MAX:6]`, but this might reduce the simulation speed of the model.
- OKB SLC data RAM and tag RAM are not supported.
- Source-based SLC cache partitioning is not supported.
- Way-based SLC cache partitioning is not supported.
- SLC/LCC partitioning dynamic allocation is not supported.
- Remote chip addresses incorrectly allocate a line in the SLC when `cache_state_modelled=true`. This can cause unexpected cache behavior that does not match the RTL, exhausting the SLC earlier or extra evictions.
- The fields in the `mxp_device_port_connect_info` register corresponding to the number of device credit slices and CAL credit slices are not supported.
- SN-F hashing in HNSAM is not supported. HNSAM SN-F hashing is a single memory range mapped to multiple SN-Fs. Example `HNFSAM_Mem_Map`:

```
0x00000001000 - 0x00000002000 SNF0 SNF1
```

The model supports mapping HN-F to a single SN-F. Example HNFSAM\_Mem\_Map:

```
0x0000001000 - 0x0000002000 SNF0
0x0000002000 - 0x0000003000 SNF1
```

- HN-Fs with different SLC sizes in the same configuration are not supported.
- Transactions to addresses unused by Device registers in the Configuration Register Space are always routed to the Configuration Space even when the SAM is configured to use those addresses.
- GIC communication over A4S ports is not supported.
- MPU with data coherency is not supported.
- MPU located in CXRH is not supported.
- No specific Debug Trace (DT) node functionality is supported. See [Plug-ins for Fast Models](#) for information about using plug-ins for trace.
- For Power State Transition related flush/disable of the SLC due to HNF\_PPU\_PWPR register, only the Operational Mode field (bits[7:4]) is considered. The Power Policy field (bits[3:0]) is ignored.
- Setting HAM mode for the Operational Mode field (bits[7:4]) of the HNF\_PPU\_PWPR is equivalent to FAM mode. The SLCH2 is neither flushed nor disabled according to the Power Policy field (bits[3:0]).
- Only hashing granularities of 4096B and above are supported.
- All RNs are assumed to have the same HN mapping. Programming RNSAMs differently between RNs is not supported.
- RNSAM cannot be programmed with the same HN-F target in a hashed and non-hashed memory region.
- Address bit masking is not supported in RNSAM or HN-F SAM.
- The OCM does not keep track of the total memory used across the S/NS security world and does not perform any eviction either. It is up to the software to make sure it does not oversubscribe the OCM.
- There is no support for RAS.
- The following limitations apply to System Cache Groups and Hash Target Groups:
  - A mix of local and remote HN-F targets are not supported.
  - CCG/CXRH target IDs in the HN target ID table are not supported (`sys_cache_grp_hn_nodeid_reg`).
  - The `hashed_target_grp_hnp_nodeid_reg` which is used for CCG/HN-P target IDs is not supported.
  - System/Hash Target Groups only support HN-Fs.
  - AXID hashing across HN-P/CCGs is not supported.
- For RN-D nodes, when software writes SYSCOREQ, DVM propagation gets enabled but SYSCOACK is not set.

## About the debug\_force\_snoop parameter

The CMN interconnect normally starts up with snooping disabled.

The parameters `rnf_sci_enable` and `rni_sci_enable` determine which connections are managed by the System Coherency Interface. Connections that are managed by the System Coherency Interface look after themselves for entering and leaving the coherency domain and `debug_force_snoop` has no effect on those ports.

However, for connections that are not managed by the System Coherency Interface, `debug_force_snoop` enables the system upstream of a port to be snooped if the upstream is not in reset and if `acchannelen_rnf` allows it.

Therefore, software does not need to know how to turn snoops in the interconnect on and off. The reset state of the upstream rnf ports is determined by the `rnf_upstream_reset_state[]` signals and must be connected for any port that could go into reset and you want managed by `debug_force_snoop`.

If the connection is managed by SCI or can never go into reset then you need not connect this and the interconnect assumes that the upstream system is not in reset and will always enable the snoops if `ACCHANNELENSx` allows it.

If you fail to connect `rnf_upstream_reset_state[]` correctly, then the upstream system might receive snoop messages while in reset and it will complain that it received a snoop request while it was in reset.

If software overrides the enables then the next change to `rnf_upstream_reset_state[]` or the interconnect reset will overwrite the software values.

Using this operation along with software controlling the enables could confuse the software. This option does not remove the responsibility of the upstream system to clean any shared dirty data from its caches before going into reset.



Note

This parameter is for debug purposes only. Do not use it instead of correctly configuring CMN.

## Iris and MTI instances for CMN600AE

This model has the following Iris instances:

Name	Instance type
CMN600AE	CMN600AE
CMN600AE.bus_slave_ocm_NS	PVBusSlave
CMN600AE.bus_slave_ocm_S	PVBusSlave
CMN600AE.cmn600_cache	PVCache
CMN600AE.cmn600_cache.downstream[Z].pvbusmaster (where Z = 0-8)	PVBusMaster

Name	Instance type
CMN600AE.cmn600_cache.upstream[Z] (where Z = 0-18)	PVBusSlave
CMN600AE.hnf_exclusive_monitor	PVBusExclusiveMonitor
CMN600AE.hnf_exclusive_monitor.bus_mapper	PVBusMapper
CMN600AE.hni_exclusive_monitorY (where Y = 0-1)	PVBusExclusiveMonitor
CMN600AE.hni_exclusive_monitorY.bus_mapper (where Y = 0-1)	PVBusMapper
CMN600AE.mpu	PVBusMapper
CMN600AE.ocm_decoder	PVBusMapper
CMN600AE.ocm_exclusive_monitor	PVBusExclusiveMonitor
CMN600AE.ocm_exclusive_monitor.bus_mapper	PVBusMapper
CMN600AE.snf_mapper	PVBusMapper

This model has the following MTI trace components:

Name	Component type
CMN600AE	CMN600AE
CMN600AE.bus_slave_ocm_NS	PVBusSlave
CMN600AE.bus_slave_ocm_S	PVBusSlave
CMN600AE.cmn600_cache	CMN600Cache
CMN600AE.cmn600_cache.downstream[Z].pvbusmaster (where Z = 0-8)	PVBusMaster
CMN600AE.cmn600_cache.upstream[Z] (where Z = 0-18)	PVBusSlave
CMN600AE.hnf_exclusive_monitor	PVBusExclusiveMonitor
CMN600AE.hnf_exclusive_monitor.bus_mapper	PVBusMapper
CMN600AE.hni_exclusive_monitorY (where Y = 0-1)	PVBusExclusiveMonitor
CMN600AE.hni_exclusive_monitorY.bus_mapper (where Y = 0-1)	PVBusMapper
CMN600AE.mpu	PVBusMapper
CMN600AE.ocm_decoder	PVBusMapper
CMN600AE.ocm_exclusive_monitor	PVBusExclusiveMonitor
CMN600AE.ocm_exclusive_monitor.bus_mapper	PVBusMapper
CMN600AE.snf_mapper	PVBusMapper

## Ports for CMN600AE

Port	Direction	Protocol	Description
event_clusters	peer	Signal	CPU event communication signal from the clusters.
event_downstream_link_signal	master	Signal	CPU event communication signal to the CMLHub. Event from the CMN towards the Hub NOTE that these are virtual "links" on underlying PCIe hardware bus.
event_upstream_link_signal	slave	Signal	Event from the Hub towards the CMN
fm_u_eri	master	Signal	FMU signal for critical errors
fm_u_fhi	master	Signal	FMU signal for non-critical errors
pvbus_m_cml_cfg	master	PVBus	CML downstream hub configuration port
pvbus_m_cml	master	PVBus	CML downstream ports

Port	Direction	Protocol	Description
pvbus_m_hni	master	PVBus	HNI downstream ports.
pvbus_m_snf	master	PVBus	SNF downstream port.
pvbus_s_apb	slave	PVBus	APB interface port.
pvbus_s_cml	slave	PVBus	CML upstream ports
pvbus_s_rnf	slave	PVBus	RNF upstream ports.
pvbus_s_rni	slave	PVBus	RNI upstream ports.
reset_in	slave	Signal	Reset signal.
rnf_sci_s	slave	SystemCoherencyInterface	System Coherency port to move the RN-F in and out of the coherency domain.
rnf_upstream_reset_in	slave	Signal	Used by the interconnect to determine the reset state of the RNF manager to check an aspect of reset sequencing. If debug_force_snoop is used, this port must be connected.
rni_sci_s	slave	SystemCoherencyInterface	System Coherency port to move the RN-D in and out of the coherency domain.
rni_upstream_reset_in	slave	Signal	Used by the interconnect to determine the reset state of the RNI manager to check an aspect of reset sequencing. If debug_force_snoop is used, this port must be connected.

## Parameters for CMN600AE

### acchannelen\_rnf



Note

DEPRECATED: Will be removed after FM 11.18. Use `rnf_sci_enable` instead.

For each `rnf` port, indicates if the port is populated with a snoop responding device or not.

The input value is a string, for example `0xffff` or `ffff`.

Type: `string`

Default value: `"0"`

### acchannelen\_rni



Note

DEPRECATED: Will be removed after FM 11.18. Use `rni_sci_enable` instead.

For each `rni` port, indicates if the port is populated with a dvm responding device or not.

The input value is a string, for example `0xffff` or `ffff`.

Type: `string`

Default value: `"0"`

### **cache\_state\_modelled**

Model the cache state.

Type: `bool`

Default value: `false`

### **debug\_force\_snoop**

The CMN600AE interconnect will normally start with snooping disabled.

The parameter `rnf_sci_enable` and `rni_sci_enable` determines which connections are managed by the System Coherency Interface.

For connections that are managed by the System Coherency Interface, then they will look after themselves for entering and leaving the coherency domain and this parameter has no effect for those ports.

However, for connections that are *not* managed by the System Coherency Interface, this parameter enables the upstream system of a port to be snooped if the upstream is not in reset and if `acchannelen_rnf` allows it.



This parameter is for debug purpose only

Do not use this parameter instead of correctly configuring CMN.

---

Type: `bool`

Default value: `false`

### **disable\_CML\_port**

If true the model won't connect the CML port when there are nodes which support the CML feature (ie CXG nodes).

Type: `bool`

Default value: `false`

### **enable\_logger**

Enable PVBUSLoggers for the downstream ports in the CMN model.

Type: `bool`

Default value: `false`

### **`enable_rnsam_to_hnf_wider_hash`**

Enable support of wider hash for the RNSAM to HNF communication.

If this variable enabled, then `bits[47:6]` from PA used in hashing function.

By default it is `[47:12]`.

Type: `bool`

Default value: `false`

### **`fdc_key`**

`por_fdc_key` register value is checked against this key.

Type: `uint8_t`

Default value: 0

### **`fmu_key`**

`por_fmu_key` register value is checked against this key.

Type: `uint8_t`

Default value: 0

### **`force_rnsam_internal`**

Force all RNSAMs to be internal independently of the mesh topology.

Type: `bool`

Default value: `true`

### **`mesh_config_file`**

Name of a file containing mesh placement of CMN600AE components.

Type: `string`

Default value: `""`

### **`number_of_mpu_programmable_regions`**

Number of MPU programmable regions.

Valid values are 0, 8, 16 and 32.

Type: `uint32_t`



Default value: 32

### **periphbase**

Value for `PERIPHBASE`. Bits [25:0] are treated 0.

Type: `uint64_t`

Default value: `0x20000000`

### **print\_cmn\_ccix\_config**

Print information about the CCIX configuration.

Type: `bool`

Default value: `false`

### **print\_cmn\_config**

Print the mesh topology and children pointers acquired from the YML file.

Type: `bool`

Default value: `false`

### **revision**

Component revision.

Currently supports r1p0.

Type: `string`

Default value: `"r1p0"`

### **rnf\_sci\_enable**

For each rnf port, indicates if the port is managed by System Coherency Interface for coherency domain entry/exit

**1**

Managed by SCI

**0**

Managed by Software

The input value is a string, for example `0xffff` or `ffff`.

Type: `string`

Default value: `"0x0"`

**rni\_sci\_enable**

For each rni port, indicates if the port is managed by System Coherency Interface for coherency domain entry/exit

**1**

Managed by SCI

**0**

Managed by Software

The input value is a string, for example `0xffff` or `ffff`.

Type: `string`

Default value: `"0x0"`

**show\_banner**

Show component banner:

**0**

supress entire banner

**1**

suppress config file

**2+**

show full banner.

Type: `uint64_t`

Default value: 2

**skip\_cmn\_config\_check**

Skip any topology configuration checks. The maximum number of devices per type not verified.

Type: `bool`

Default value: `false`

**use\_yaml\_periphbase**

Use yaml param `CFGM_PERIPHBASE_PARAM` to specify periphbase address.

If false, model parameter `periphbase` will be used.

Type: `bool`

Default value: `false`

## 3.140 CMN600CMLHub

Defined in `LISA/CMN600CMLHub.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
N/A	Preliminary support

For an explanation of the quality levels, see [Quality level definitions](#).

### About CMN600CMLHub

CMN600 CML Interconnect Hub Fast Model.

### Iris and MTI instances for CMN600CMLHub

This model has the following Iris instances:

Name	Instance type
CMN600CMLHub	CMN600CMLHub
CMN600CMLHub.CMN600CMLHubCache	PVCache
CMN600CMLHub.CMN600CMLHubCache.downstream[Z].pvbusmaster (where Z = 0–4)	PVBusMaster
CMN600CMLHub.CMN600CMLHubCache.upstream[Z] (where Z = 0–3)	PVBusSlave
CMN600CMLHub.bus_s_cfg	PVBusSlave
CMN600CMLHub.cache_downstream_exclusive_monitorY (where Y = 0–3)	PVBusExclusiveMonitor
CMN600CMLHub.cache_downstream_exclusive_monitorY.bus_mapper (where Y = 0–3)	PVBusMapper

This model has the following MTI trace components:

Name	Component type
CMN600CMLHub	CMN600CML
CMN600CMLHub.CMN600CMLHubCache	PVCache
CMN600CMLHub.CMN600CMLHubCache.downstream[Z].pvbusmaster (where Z = 0–4)	PVBusMaster
CMN600CMLHub.CMN600CMLHubCache.upstream[Z] (where Z = 0–3)	PVBusSlave
CMN600CMLHub.bus_s_cfg	PVBusSlave
CMN600CMLHub.cache_downstream_exclusive_monitorY (where Y = 0–3)	PVBusExclusiveMonitor
CMN600CMLHub.cache_downstream_exclusive_monitorY.bus_mapper (where Y = 0–3)	PVBusMapper

### Ports for CMN600CMLHub

Port	Direction	Protocol	Description
event_downstream_link_signal	slave	Signal	CPU downstream event communication signal.
event_upstream_link_signal	master	Signal	CPU upstream event communication signal.
pvbus_m	master	PVBus	Downstream CCIX port.
pvbus_s_cfg	slave	PVBus	Upstream config ports.
pvbus_s	slave	PVBus	Upstream CCIX ports.

Port	Direction	Protocol	Description
reset_signal	slave	Signal	Reset signal.

## Parameters for CMN600CMLHub

### **cache\_state\_modelled**

Model the cache state.

Type: `bool`

Default value: `false`

### **enable\_logger**

Enable PVBUSLoggers for the downstream ports in the CMN model.

Type: `bool`

Default value: `false`

## 3.141 CMN650

Defined in `LISA/CMN650.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r1p1	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### About CMN650

- Major IP revisions (rX) are modeled and are controlled by the `revision` parameter.
- Minor IP revisions (pY) are not modeled. There are no functional differences in the model between pY revisions, with the exception of the `periph_id2` register.
- The topology YAML 'version' field overrides the model parameter 'revision' when selecting the CMN revision.
- To configure the model, you must have installed Arm Socrates. The `mesh_config_file` parameter defines the mesh placement of the CHI nodes. Set it to the name of the yml configuration file emitted by the Socrates export process. Fast Models requires the configuration file to pass Design Rule Check (DRC) in Socrates and does not support manually editing the configuration file. You must use version r1p4-01rel0 or later of Socrates to generate the file. For more information about Socrates, see [Arm Socrates](#) on Arm Developer. To download it, contact Arm Technical Support.
- Interconnect models are based on the TRM description and do not typically model RTL defects.

The model supports the following features:

- rnf, rni/rnd, hni, and snf/sbsx interface ports. The mapping between the port number and `NodeId` is based on the `NodeId` index. For example, RNF2 controls `pvbuss_s_rnf[2]`. Its index is specified in the `node_info` register as `logical_id`. Similar behavior can be expected for HN-I.

If both RN-D and RN-I nodes are present, then all starting rni ports are mapped to RN-D nodes and then the RN-I nodes. For example, for CMN650 with two RN-D nodes, one RN-I node, and given each RN-I or RN-D node controls three interface ports, `pvbuss_s_rni[0-2]` maps to RND0, `pvbuss_s_rni[3-5]` maps to RND1 and `pvbuss_s_rni[6-8]` maps to RNI0.

Similarly, SN-F and SBSX nodes are mapped to `pvbuss_m_snf[]` ports where starting ports are mapped to SN-F and then SBSX nodes.

- Mapping HN\*\_SLC\_SIZE\_PARAM values to cache sizes:

-8

OKB where `HN*_SLC_NUM_WAYS_PARAM=16`



This value is not supported by the model.

---

-2

128KB where `HN*_SLC_NUM_WAYS_PARAM=16`

-1

256KB where `HN*_SLC_NUM_WAYS_PARAM=16`

0

512KB where `HN*_SLC_NUM_WAYS_PARAM=16`

1

1MB where `HN*_SLC_NUM_WAYS_PARAM=16`

2

2MB where `HN*_SLC_NUM_WAYS_PARAM=16`

3

3MB where `HN*_SLC_NUM_WAYS_PARAM=12`

3

4MB where `HN*_SLC_NUM_WAYS_PARAM=16`

- The Address Based Flush (ABF) feature is available. After it has started, the ABF completes instantly. The ABF implementation has the following limitations:
  - In the Fast Model, any HN-F in the interconnect can perform an ABF on any address range even if that range is not part of the System Cache Group (SCG) that the HN-F belongs to.
  - Clearing the OCM with an ABF is not supported. A trace has been added to highlight this.

- When `HN_F_AB_F_PR.abf_mode` is Reserved and `HN_F_AB_F_PR.abf_enable` is set, an error is printed and the ABF status register is set to complete successfully, but no operation takes place.
- Because an ABF completes instantly, the `SF_PM_TRANSITION_ABORT` state cannot be reached. Contact Arm Technical Support with questions about support for this error condition.
- The optional interrupt `INTREQ_PPU` is not supported.

## Limitations

- Out of scope:
  - PMU counters are not supported. Counter registers are implemented as **RAZ**.
  - QoS is not supported and all related registers are **RAZ/WI**.
  - Error injection and error generation are not supported. All error registers are **RAZ/WI**.
  - Power, clock, and interrupt signals are not supported.

The P-channel (power) signals `PREQ_LOGIC`, `PSTATE_LOGIC`, `PACCEPT_LOGIC`, `PDENY_LOGIC`, and `PACTIVE_LOGIC`, are not implemented. The model behaves as if `PSTATE_LOGIC` is 5'b00000. So the initial power state of the HN-F nodes is NOSFSLC/OFF. This is reflected in the reset values of `cmn_hns_ppu_pwpr`, which was named `por_hnf_ppu_pwpr` in earlier IP revisions. Software must write the `*_ppu_pwpr` registers to change the power state. Both HAM and FAM power states enable all of the SLC ways.

- Protocol credits and flit buffers are not supported.
- Snoop filtering is not supported.
- Prefetch Target operations are not supported.
- Non-XY routing behavior is not supported.
- The MTE feature has been minimally tested and the functionality cannot be guaranteed. MTE error injection and detection are not supported.
- An access to a reserved or nonexistent register does not abort. It returns `pv::Tx_Data::TX_OK` and is **RAZ/WI**.
- Topologies with three I/O masters connected through an RN-I bridge device have been tested. Configurations where multiple I/O masters share a single ACE-lite port have not been tested.
- RNSAMs external to the mesh network are not functionally supported.



Note

Model parameter `force_rnsam_internal` is true by default. If this parameter is set to false and the topology has an RNFx (non-ESAM), the external bit (bit[31]) of the RNSAM children pointer is enabled. This bit is the only functionality governed by this parameter.

- By default, HN-F hashing uses the address [MAX:12] instead of the actual address [MAX:6], due to the DMI mechanism in the model. Enable the parameter `enable_rnsam_to_hnf_wider_hash` to make hashing logic use the actual address [MAX:6], but this might reduce the simulation speed of the model.
- OKB SLC data RAM and tag RAM are not supported.

- Source-based SLC cache partitioning is not supported.
- Way-based SLC cache partitioning is not supported.
- SLC/LCC partitioning dynamic allocation is not supported.
- Remote chip addresses incorrectly allocate a line in the SLC when `cache_state_modelled=true`. This can cause unexpected cache behavior that does not match the RTL, exhausting the SLC earlier or extra evictions.
- The fields in the `mxp_device_port_connect_info` register corresponding to the number of device credit slices and the number of CAL credit slices are not supported.
- SN-F hashing in HNSAM is not supported. HNSAM SN-F hashing is a single memory range mapped to multiple SN-Fs. Example HNFSAM\_Mem\_Map:

```
0x00000001000 - 0x00000002000 SNF0 SNF1
```

The model supports mapping HN-F to a single SN-F. Example HNFSAM\_Mem\_Map:

```
0x00000001000 - 0x00000002000 SNF0
0x00000002000 - 0x00000003000 SNF1
```

- HN-Fs with different SLC sizes in the same configuration are not supported.
- Transactions to addresses unused by Device registers in the Configuration Register Space are always routed to the Configuration Space even when the SAM is configured to use those addresses.
- GIC communication over A4S ports is not supported.
- MPAM features are not supported in the model, and the configuration-dependent register reset values do not match the RTL. Some of the registers (`por_hnf_mpam_idr` to `por_hnf_mpam_mbwumon_idr`) shared between security modes are not present in MPAM\_S.
- The following limitations apply to System Cache Groups and Hash Target Groups:
  - A mix of local and remote HN-F targets is not supported.
  - CCG/CXRH target IDs in the HN target ID table are not supported (`sys_cache_grp_hn_nodeid_reg`).
  - The `hashed_target_grp_hnp_nodeid_reg` which is used for CCG/HN-P target IDs is not supported.
  - System/Hash Target Groups only support HN-Fs.
  - AXID hashing across HN-P/CCGs is not supported.
- The MPAM reset values described in `por_hnf_mpam_ns_por_hnf_mpam_idr[31:24]` default to the not-supported state. The model parameter `hnf_mpam_idr_override` can be used to override this but does not implement any of the functionality.
- MPAM\_S `secure_register_groups_override` is not supported.
- No specific Debug Trace (DT) node functionality is supported. See [Plug-ins for Fast Models](#) for information about using plug-ins for trace.

- For Power State Transition related flush/disable of the SLC due to HNF\_PPU\_PWPR register, only the Operational Mode field (bits[7:4]) is considered. The Power Policy field (bits[3:0]) is ignored.
- Setting HAM mode for the Operational Mode field (bits[7:4]) of the HNF\_PPU\_PWPR is equivalent to FAM mode. The SLCH2 is neither flushed nor disabled according to the Power Policy field (bits[3:0]).
- Only hashing granularities of 4096B and above are supported.
- The OCM does not keep track of the total memory used across the S/NS security world and does not perform any eviction either. It is up to the software to make sure it does not oversubscribe the OCM.
- All RNs are assumed to have the same HN mapping. Programming RNSAMs differently between RNs is not supported.
- RNSAM cannot be programmed with the same HN-F target in a hashed and non-hashed memory region.
- `RNSAM_STATUS.default_target_type` and `RNSAM_STATUS.default_nodeid` are used as the default target only when `RNSAM_STATUS.use_default_node=1`. Otherwise, when a txn is sent into the model that does not belong to any of the address regions programmed in the RNSAM, it is routed to the HN-D irrespective of what is programmed in the `RNSAM_STATUS.default_target_type` and `RNSAM_STATUS.default_nodeid` fields.
- Address bit masking is not supported in RNSAM or HN-F SAM.
- HN-F SAM limitations:
  - Address masking in default hash regions in HN-F SAM is not supported.
  - Hashing in default hash regions in HN-F SAM is not supported. However, trace `SNF_HASHING_Target_SNF` displays the target SNF as if 3SN/6SN hashing were enabled.



The transaction is always routed to the SNO nodeid programmed in the SAM\_CONTROL register.

- 
- Re-programming regions in HN-F SAM is not tested.
  - Hashing across CCGs in HN-F SAM is not supported.
  - For an HTG/SCG, the model does not consider the secondary region if the primary region is not valid.
  - There is no support for RAS.
  - For RN-D nodes, when software writes SYSCOREQ, DVM propagation gets enabled but SYSCOACK is not set.

### About the `debug_force_snoop` parameter

The CMN interconnect normally starts up with snooping disabled.

The parameters `rnf_sci_enable` and `rni_sci_enable` determine which connections are managed by the System Coherency Interface. Connections that are managed by the System



Coherency Interface look after themselves for entering and leaving the coherency domain and `debug_force_snoop` has no effect on those ports.

However, for connections that are not managed by the System Coherency Interface, `debug_force_snoop` enables the system upstream of a port to be snooped if the upstream is not in reset and if `acchannelen_rnf` allows it.

Therefore, software does not need to know how to turn snoops in the interconnect on and off. The reset state of the upstream rnf ports is determined by the `rnf_upstream_reset_state[]` signals and must be connected for any port that could go into reset and you want managed by `debug_force_snoop`.

If the connection is managed by SCI or can never go into reset then you need not connect this and the interconnect assumes that the upstream system is not in reset and will always enable the snoops if `ACCHANNELENSx` allows it.

If you fail to connect `rnf_upstream_reset_state[]` correctly, then the upstream system might receive snoop messages while in reset and it will complain that it received a snoop request while it was in reset.

If software overrides the enables then the next change to `rnf_upstream_reset_state[]` or the interconnect reset will overwrite the software values.

Using this operation along with software controlling the enables could confuse the software. This option does not remove the responsibility of the upstream system to clean any shared dirty data from its caches before going into reset.



This parameter is for debug purposes only. Do not use it instead of correctly configuring CMN.

## Iris and MTI instances for CMN650

This model has the following Iris instances:

Name	Instance type
CMN650	CMN650
CMN650.bus_slave_ocm_NS	PVBusSlave
CMN650.bus_slave_ocm_S	PVBusSlave
CMN650.cmn600_cache	PVCache
CMN650.cmn600_cache.downstream[Z].pvbusmaster (where Z = 0-10)	PVBusMaster
CMN650.cmn600_cache.upstream[Z] (where Z = 0-117)	PVBusSlave
CMN650.hnf_exclusive_monitor	PVBusExclusiveMonitor
CMN650.hnf_exclusive_monitor.bus_mapper	PVBusMapper
CMN650.hni_exclusive_monitorY (where Y = 0-3)	PVBusExclusiveMonitor
CMN650.hni_exclusive_monitorY.bus_mapper (where Y = 0-3)	PVBusMapper

Name	Instance type
CMN650.ocm_decoder	PVBusMapper
CMN650.ocm_exclusive_monitor	PVBusExclusiveMonitor
CMN650.ocm_exclusive_monitor.bus_mapper	PVBusMapper
CMN650.snf_mapper	PVBusMapper

This model has the following MTI trace components:

Name	Component type
CMN650	CMN650
CMN650.bus_slave_ocm_NS	PVBusSlave
CMN650.bus_slave_ocm_S	PVBusSlave
CMN650.cmn600_cache	CMN600Cache
CMN650.cmn600_cache.downstream[Z].pvbusmaster (where Z = 0-10)	PVBusMaster
CMN650.cmn600_cache.upstream[Z] (where Z = 0-117)	PVBusSlave
CMN650.hnf_exclusive_monitor	PVBusExclusiveMonitor
CMN650.hnf_exclusive_monitor.bus_mapper	PVBusMapper
CMN650.hni_exclusive_monitorY (where Y = 0-3)	PVBusExclusiveMonitor
CMN650.hni_exclusive_monitorY.bus_mapper (where Y = 0-3)	PVBusMapper
CMN650.ocm_decoder	PVBusMapper
CMN650.ocm_exclusive_monitor	PVBusExclusiveMonitor
CMN650.ocm_exclusive_monitor.bus_mapper	PVBusMapper
CMN650.snf_mapper	PVBusMapper

## Ports for CMN650

Port	Direction	Protocol	Description
event_clusters	peer	Signal	CPU event communication signal from the clusters.
event_downstream_link_signal	master	Signal	CPU event communication signal to the CMLHub. Event from the CMN towards the Hub NOTE that these are virtual "links" on underlying PCIe hardware bus.
event_upstream_link_signal	slave	Signal	Event from the Hub towards the CMN
pvbus_m_cml_cfg	master	PVBus	CML downstream hub configuration port
pvbus_m_cml	master	PVBus	CML downstream ports
pvbus_m_hni	master	PVBus	HNI downstream ports.
pvbus_m_snf	master	PVBus	SNF downstream port.
pvbus_s_apb	slave	PVBus	APB interface port.
pvbus_s_cml	slave	PVBus	CML upstream ports
pvbus_s_rnf	slave	PVBus	RNF upstream ports.
pvbus_s_rni	slave	PVBus	RNI upstream ports.
reset_in	slave	Signal	Reset signal.
rnf_sci_s	slave	SystemCoherencyInterface	System Coherency port to move the RN-F in and out of the coherency domain.

Port	Direction	Protocol	Description
rnf_upstream_reset_in	slave	Signal	Used by the interconnect to determine the reset state of the RNF manager to check an aspect of reset sequencing. If debug_force_snoop is used, this port must be connected.
rni_sci_s	slave	SystemCoherencyInterface	System Coherency port to move the RN-D in and out of the coherency domain.
rni_upstream_reset_in	slave	Signal	Used by the interconnect to determine the reset state of the RNI manager to check an aspect of reset sequencing. If debug_force_snoop is used, this port must be connected.

## Parameters for CMN650

### acchannelen\_rnf



DEPRECATED: Will be removed after FM 11.18. Use `rnf_sci_enable` instead.

For each `rnf` port, indicates if the port is populated with a snoop responding device or not.

The input value is a string, for example `0xffff` or `ffff`.

Type: `string`

Default value: `"0"`

### acchannelen\_rni



DEPRECATED: Will be removed after FM 11.18. Use `rni_sci_enable` instead.

For each `rni` port, indicates if the port is populated with a dvm responding device or not.

The input value is a string, for example `0xffff` or `ffff`.

Type: `string`

Default value: `"0"`

### cache\_state\_modelled

Model the cache state.

Type: `bool`

Default value: `false`

### **debug\_force\_snoop**

The CMN650 interconnect will normally start with snooping disabled.

The parameter `rnf_sci_enable` and `rni_sci_enable` determines which connections are managed by the System Coherency Interface.

For connections that are managed by the System Coherency Interface, then they will look after themselves for entering and leaving the coherency domain and this parameter has no effect for those ports.

However, for connections that are *not* managed by the System Coherency Interface, this parameter enables the upstream system of a port to be snooped if the upstream is not in reset and if `acchannelen_rnf` allows it.



Note

This parameter is for debug purpose only

Do not use this parameter instead of correctly configuring CMN.

---

Type: `bool`

Default value: `false`

### **disable\_CML\_port**

If true the model won't connect the CML port when there are nodes which support the CML feature (ie CXG nodes).

Type: `bool`

Default value: `false`

### **enable\_logger**

Enable PVBUSLoggers for the downstream ports in the CMN model.

Type: `bool`

Default value: `false`

### **enable\_rnsam\_to\_hnf\_wider\_hash**

Enable support of wider hash for the RNSAM to HNF communication.

If this variable enabled, then `bits[47:6]` from PA used in hashing function.

By default it is `[47:12]`.

Type: `bool`

Default value: `false`

**force\_rnsam\_internal**

Force all RNSAMs to be internal independently of the mesh topology.

Type: `bool`

Default value: `true`

**hnf\_mpam\_idr\_override**

Set to override `hnf_mpam_idr[31:24]` value. Bit[28] is Reserved and is ignored.

Type: `uint64_t`

Default value: `0`

**mesh\_config\_file**

Name of a file containing mesh placement of CMN650 components.

Type: `string`

Default value: `""`

**periphbase**

Value for `PERIPHBASE`. Bits [27:0] are treated 0.

Type: `uint64_t`

Default value: `0x20000000`

**print\_cmn\_ccix\_config**

Print information about the CCI configuration.

Type: `bool`

Default value: `false`

**print\_cmn\_config**

Print the mesh topology and children pointers acquired from the YML file.

Type: `bool`

Default value: `false`

**revision**

Component revision.

Currently supports r1p1.

Type: `string`

Default value: `"r1p1"`

### **`rnf_sci_enable`**

For each rnf port, indicates if the port is managed by System Coherency Interface for coherency domain entry/exit

**1**

Managed by SCI

**0**

Managed by Software

The input value is a string, for example `0xffff` or `ffff`.

Type: `string`

Default value: `"0x0"`

### **`rni_sci_enable`**

For each rni port, indicates if the port is managed by System Coherency Interface for coherency domain entry/exit

**1**

Managed by SCI

**0**

Managed by Software

The input value is a string, for example `0xffff` or `ffff`.

Type: `string`

Default value: `"0x0"`

### **`show_banner`**

Show component banner:

**0**

supress entire banner

**1**

suppress config file

**2+**

show full banner.

Type: `uint64_t`

Default value: 2

**skip\_cmn\_config\_check**

Skip any topology configuration checks. The maximum number of devices per type not verified.

Type: `bool`

Default value: `false`

**use\_yaml\_periphbase**

Use yaml param `CFGM_PERIPHBASE_PARAM` to specify periphbase address.

If false, model parameter `periphbase` will be used.

Type: `bool`

Default value: `false`

## 3.142 CMN650R2

Defined in `LISA/CMN650R2.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r2p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### About CMN650R2

- Major IP revisions (rX) are modeled and are controlled by the `revision` parameter.
- Minor IP revisions (pY) are not modeled. There are no functional differences in the model between pY revisions, with the exception of the `periph_id2` register.
- The topology YAML 'version' field overrides the model parameter 'revision' when selecting the CMN revision.
- To configure the model, you must have installed Arm Socrates. The `mesh_config_file` parameter defines the mesh placement of the CHI nodes. Set it to the name of the yaml configuration file emitted by the Socrates export process. Fast Models requires the configuration file to pass Design Rule Check (DRC) in Socrates and does not support manually editing the configuration file. You must use version r1p4-01rel0 or later of Socrates to generate the file. For more information about Socrates, see [Arm Socrates](#) on Arm Developer. To download it, contact Arm Technical Support.
- Interconnect models are based on the TRM description and do not typically model RTL defects.

The model supports the following features:

- rnf, rni/rnd, hni, and snf/sbsx interface ports. The mapping between the port number and `nodeId` is based on the `nodeId` index. For example, RNF2 controls `pvbus_s_rnf[2]`. Its index is specified in the `node_info` register as `logical_id`. Similar behavior can be expected for HN-I.

If both RN-D and RN-I nodes are present, then all starting rni ports are mapped to RN-D nodes and then the RN-I nodes. For example, for CMN650 with two RN-D nodes, one RN-I node, and given each RN-I or RN-D node controls three interface ports, `pvbus_s_rni[0-2]` maps to RND0, `pvbus_s_rni[3-5]` maps to RND1 and `pvbus_s_rni[6-8]` maps to RNI0.

Similarly, SN-F and SBSX nodes are mapped to `pvbus_m_snf[]` ports where starting ports are mapped to SN-F and then SBSX nodes.

- Mapping `HN*_SLC_SIZE_PARAM` values to cache sizes:

-8

0KB where `HN*_SLC_NUM_WAYS_PARAM=16`



Note

This value is not supported by the model.

-2

128KB where `HN*_SLC_NUM_WAYS_PARAM=16`

-1

256KB where `HN*_SLC_NUM_WAYS_PARAM=16`

0

512KB where `HN*_SLC_NUM_WAYS_PARAM=16`

1

1MB where `HN*_SLC_NUM_WAYS_PARAM=16`

2

2MB where `HN*_SLC_NUM_WAYS_PARAM=16`

3

3MB where `HN*_SLC_NUM_WAYS_PARAM=12`

3

4MB where `HN*_SLC_NUM_WAYS_PARAM=16`

- The Address Based Flush (ABF) feature is available. After it has started, the ABF completes instantly. The ABF implementation has the following limitations:
  - In the Fast Model, any HN-F in the interconnect can perform an ABF on any address range even if that range is not part of the System Cache Group (SCG) that the HN-F belongs to.
  - Clearing the OCM with an ABF is not supported. A trace has been added to highlight this.
  - When `HNF_ABF_PR.abf_mode` is Reserved and `HNF_ABF_PR.abf_enable` is set, an error is printed and the ABF status register is set to complete successfully, but no operation takes place.



- Because an ABF completes instantly, the `SF_PM_TRANSITION_ABORT` state cannot be reached. Contact Arm Technical Support with questions about support for this error condition.
- The optional interrupt `INTREQPPU` is not supported.

## Limitations

- Out of scope:
  - PMU counters are not supported. Counter registers are implemented as **RAZ**.
  - QoS is not supported and all related registers are **RAZ/WI**.
  - Error injection and error generation are not supported. All error registers are **RAZ/WI**.
  - Power, clock, and interrupt signals are not supported.

The P-channel (power) signals `PREQ_LOGIC`, `PSTATE_LOGIC`, `PACCEPT_LOGIC`, `PDENY_LOGIC`, and `PACTIVE_LOGIC`, are not implemented. The model behaves as if `PSTATE_LOGIC` is 5'b00000. So the initial power state of the HN-F nodes is `NOSFSLC/OFF`. This is reflected in the reset values of `cmn_hns_ppu_pwpr`, which was named `por_hnf_ppu_pwpr` in earlier IP revisions. Software must write the `*_ppu_pwpr` registers to change the power state. Both HAM and FAM power states enable all of the SLC ways.

- Protocol credits and flit buffers are not supported.
- Snoop filtering is not supported.
- Prefetch Target operations are not supported.
- Non-XY routing behavior is not supported.
- The MTE feature has been minimally tested and the functionality cannot be guaranteed. MTE error injection and detection are not supported.
- An access to a reserved or nonexistent register does not abort. It returns `pv::Tx_Data::TX_OK` and is **RAZ/WI**.
- Topologies with three I/O masters connected through an RN-I bridge device have been tested. Configurations where multiple I/O masters share a single ACE-lite port have not been tested.
- RNSAMs external to the mesh network are not functionally supported.



Note

Model parameter `force_rnsam_internal` is true by default. If this parameter is set to false and the topology has an RNFx (non-ESAM), the external bit (bit[31]) of the RNSAM children pointer is enabled. This bit is the only functionality governed by this parameter.

- By default, HN-F hashing uses the address `[MAX:12]` instead of the actual address `[MAX:6]`, due to the DMI mechanism in the model. Enable the parameter `enable_rnsam_to_hnf_wider_hash` to make hashing logic use the actual address `[MAX:6]`, but this might reduce the simulation speed of the model.
- OKB SLC data RAM and tag RAM are not supported.
- Source-based SLC cache partitioning is not supported.
- Way-based SLC cache partitioning is not supported.

- SLC/LCC partitioning dynamic allocation is not supported.
- Remote chip addresses incorrectly allocate a line in the SLC when `cache_state_modelled=true`. This can cause unexpected cache behavior that does not match the RTL, exhausting the SLC earlier or extra evictions.
- The fields in the `mvp_device_port_connect_info` register corresponding to the number of device credit slices and the number of CAL credit slices are not supported.
- SN-F hashing in HNSAM is not supported. HNSAM SN-F hashing is a single memory range mapped to multiple SN-Fs. Example HNFSAM\_Mem\_Map:

```
0x00000001000 - 0x00000002000 SNF0 SNF1
```

The model supports mapping HN-F to a single SN-F. Example HNFSAM\_Mem\_Map:

```
0x00000001000 - 0x00000002000 SNF0
0x00000002000 - 0x00000003000 SNF1
```

- HN-Fs with different SLC sizes in the same configuration are not supported.
- Transactions to addresses unused by Device registers in the Configuration Register Space are always routed to the Configuration Space even when the SAM is configured to use those addresses.
- GIC communication over A4S ports is not supported.
- MPAM features are not supported in the model, and the configuration-dependent register reset values do not match the RTL. Some of the registers (`por_hnf_mpam_idr` to `por_hnf_mpam_mbwumon_idr`) shared between security modes are not present in MPAM\_S.
- The following limitations apply to System Cache Groups and Hash Target Groups:
  - A mix of local and remote HN-F targets is not supported.
  - CCG/CXRH target IDs in the HN target ID table are not supported (`sys_cache_grp_hn_nodeid_reg`).
  - The `hashed_target_grp_hnp_nodeid_reg` which is used for CCG/HN-P target IDs is not supported.
  - System/Hash Target Groups only support HN-Fs.
  - AXID hashing across HN-P/CCGs is not supported.
- The MPAM reset values described in `por_hnf_mpam_ns_por_hnf_mpam_idr[31:24]` default to the not-supported state. The model parameter `hnf_mpam_idr_override` can be used to override this but does not implement any of the functionality.
- MPAM\_S `secure_register_groups_override` is not supported.
- No specific Debug Trace (DT) node functionality is supported. See [Plug-ins for Fast Models](#) for information about using plug-ins for trace.
- For Power State Transition related flush/disable of the SLC due to HNF\_PPU\_PWPR register, only the Operational Mode field (bits[7:4]) is considered. The Power Policy field (bits[3:0]) is ignored.

- Setting HAM mode for the Operational Mode field (bits[7:4]) of the HNF\_PPU\_PWPR is equivalent to FAM mode. The SLCH2 is neither flushed nor disabled according to the Power Policy field (bits[3:0]).
- Only hashing granularities of 4096B and above are supported.
- The OCM does not keep track of the total memory used across the S/NS security world and does not perform any eviction either. It is up to the software to make sure it does not oversubscribe the OCM.
- All RNs are assumed to have the same HN mapping. Programming RNSAMs differently between RNs is not supported.
- RNSAM cannot be programmed with the same HN-F target in a hashed and non-hashed memory region.
- `RNSAM_STATUS.default_target_type` and `RNSAM_STATUS.default_nodeid` are used as the default target only when `RNSAM_STATUS.use_default_node=1`. Otherwise, when a txn is sent into the model that does not belong to any of the address regions programmed in the RNSAM, it is routed to the HN-D irrespective of what is programmed in the `RNSAM_STATUS.default_target_type` and `RNSAM_STATUS.default_nodeid` fields.
- Address bit masking is not supported in RNSAM or HN-F SAM.
- HN-F SAM limitations:
  - Address masking in default hash regions in HN-F SAM is not supported.
  - Hashing in default hash regions in HN-F SAM is not supported. However, trace `SNF_HASHING_Target_SNF` displays the target SNF as if 3SN/6SN hashing were enabled.



The transaction is always routed to the SN0 nodeid programmed in the `SAM_CONTROL` register.

- 
- Re-programming regions in HN-F SAM is not tested.
  - Hashing across CCGs in HN-F SAM is not supported.
  - CAL2 support for HN-P and RN-D is not tested.
  - The maximum number of 256 RN-Fs is not verified. 74 is the largest number tested.
  - The maximum number of 40 SNs is not verified. 20 is the largest number tested.
  - The maximum number of 36 RN-Is is not verified. 16 is the largest number tested.
  - The maximum number of 16 HN-Is is not verified. 5 is the largest number tested.
  - Early DVM completion is not supported.
  - CCIX port to port forwarding is not supported.
  - No support for up to 512 CXRAs with no RAID aliasing and 256 RN-Fs on a single chip.
  - Each HN-F can support tracking of up to 512 logical processors for exclusive operations. However, the value of the RO field `num_excl` in the HN-F unit info register cannot exceed 255.
  - For RN-D nodes, when software writes `SYSSCOREQ`, DVM propagation gets enabled but `SYSOCOACK` is not set.

## About the debug\_force\_snoop parameter

The CMN interconnect normally starts up with snooping disabled.

The parameters `rnf_sci_enable` and `rni_sci_enable` determine which connections are managed by the System Coherency Interface. Connections that are managed by the System Coherency Interface look after themselves for entering and leaving the coherency domain and `debug_force_snoop` has no effect on those ports.

However, for connections that are not managed by the System Coherency Interface, `debug_force_snoop` enables the system upstream of a port to be snooped if the upstream is not in reset and if `acchannelen_rnf` allows it.

Therefore, software does not need to know how to turn snoops in the interconnect on and off. The reset state of the upstream rnf ports is determined by the `rnf_upstream_reset_state[]` signals and must be connected for any port that could go into reset and you want managed by `debug_force_snoop`.

If the connection is managed by SCI or can never go into reset then you need not connect this and the interconnect assumes that the upstream system is not in reset and will always enable the snoops if `ACCHANNELENSx` allows it.

If you fail to connect `rnf_upstream_reset_state[]` correctly, then the upstream system might receive snoop messages while in reset and it will complain that it received a snoop request while it was in reset.

If software overrides the enables then the next change to `rnf_upstream_reset_state[]` or the interconnect reset will overwrite the software values.

Using this operation along with software controlling the enables could confuse the software. This option does not remove the responsibility of the upstream system to clean any shared dirty data from its caches before going into reset.



Note

This parameter is for debug purposes only. Do not use it instead of correctly configuring CMN.

## Iris and MTI instances for CMN650R2

This model has the following Iris instances:

Name	Instance type
CMN650R2	CMN650R2
CMN650R2.bus_slave_ocm_NS	PVBusSlave
CMN650R2.bus_slave_ocm_S	PVBusSlave
CMN650R2.cmn600_cache	PVCache
CMN650R2.cmn600_cache.downstream[U].pvbusmaster (where U = 0-9)	PVBusMaster

Name	Instance type
CMN650R2.cmn600_cache.upstream[U] (where U = 0-33)	PVBusSlave
CMN650R2.hnf_exclusive_monitor	PVBusExclusiveMonitor
CMN650R2.hnf_exclusive_monitor.bus_mapper	PVBusMapper
CMN650R2.hni_exclusive_monitorZ (where Z = 0-2)	PVBusExclusiveMonitor
CMN650R2.hni_exclusive_monitorZ.bus_mapper (where Z = 0-2)	PVBusMapper
CMN650R2.ocm_decoder	PVBusMapper
CMN650R2.ocm_exclusive_monitor	PVBusExclusiveMonitor
CMN650R2.ocm_exclusive_monitor.bus_mapper	PVBusMapper
CMN650R2.snf_mapper	PVBusMapper

This model has the following MTI trace components:

Name	Component type
CMN650R2	CMN650R2
CMN650R2.bus_slave_ocm_NS	PVBusSlave
CMN650R2.bus_slave_ocm_S	PVBusSlave
CMN650R2.cmn600_cache	CMN600Cache
CMN650R2.cmn600_cache.downstream[U].pvbusmaster (where U = 0-9)	PVBusMaster
CMN650R2.cmn600_cache.upstream[U] (where U = 0-33)	PVBusSlave
CMN650R2.hnf_exclusive_monitor	PVBusExclusiveMonitor
CMN650R2.hnf_exclusive_monitor.bus_mapper	PVBusMapper
CMN650R2.hni_exclusive_monitorZ (where Z = 0-2)	PVBusExclusiveMonitor
CMN650R2.hni_exclusive_monitorZ.bus_mapper (where Z = 0-2)	PVBusMapper
CMN650R2.ocm_decoder	PVBusMapper
CMN650R2.ocm_exclusive_monitor	PVBusExclusiveMonitor
CMN650R2.ocm_exclusive_monitor.bus_mapper	PVBusMapper
CMN650R2.snf_mapper	PVBusMapper

## Ports for CMN650R2

Port	Direction	Protocol	Description
event_clusters	peer	Signal	CPU event communication signal from the clusters.
event_downstream_link_signal	master	Signal	CPU event communication signal to the CMLHub. Event from the CMN towards the Hub NOTE that these are virtual "links" on underlying PCIe hardware bus.
event_upstream_link_signal	slave	Signal	Event from the Hub towards the CMN
pvbus_m_cml_cfg	master	PVBus	CML downstream hub configuration port
pvbus_m_cml	master	PVBus	CML downstream ports
pvbus_m_hni	master	PVBus	HNI downstream ports.
pvbus_m_snf	master	PVBus	SNF downstream port.
pvbus_s_apb	slave	PVBus	APB interface port.
pvbus_s_cml	slave	PVBus	CML upstream ports

Port	Direction	Protocol	Description
pvbus_s_rnf	slave	PVBus	RNF upstream ports.
pvbus_s_rni	slave	PVBus	RNI upstream ports.
reset_in	slave	Signal	Reset signal.
rnf_sci_s	slave	SystemCoherencyInterface	System Coherency port to move the RN-F in and out of the coherency domain.
rnf_upstream_reset_in	slave	Signal	Used by the interconnect to determine the reset state of the RNF manager to check an aspect of reset sequencing. If debug_force_snoop is used, this port must be connected.
rni_sci_s	slave	SystemCoherencyInterface	System Coherency port to move the RN-D in and out of the coherency domain.
rni_upstream_reset_in	slave	Signal	Used by the interconnect to determine the reset state of the RNI manager to check an aspect of reset sequencing. If debug_force_snoop is used, this port must be connected.

## Parameters for CMN650R2

### acchannelen\_rnf



Note

DEPRECATED: Will be removed after FM 11.18. Use `rnf_sci_enable` instead.

For each `rnf` port, indicates if the port is populated with a snoop responding device or not.

The input value is a string, for example `0xffff` or `ffff`.

Type: `string`

Default value: `"0"`

### acchannelen\_rni



Note

DEPRECATED: Will be removed after FM 11.18. Use `rni_sci_enable` instead.

For each `rni` port, indicates if the port is populated with a dvm responding device or not.

The input value is a string, for example `0xffff` or `ffff`.

Type: `string`

Default value: `"0"`

**cache\_state\_modelled**

Model the cache state.

Type: `bool`

Default value: `false`

**debug\_force\_snoop**

The CMN650R2 interconnect will normally start with snooping disabled.

The parameter `rnf_sci_enable` and `rni_sci_enable` determines which connections are managed by the System Coherency Interface.

For connections that are managed by the System Coherency Interface, then they will look after themselves for entering and leaving the coherency domain and this parameter has no effect for those ports.

However, for connections that are *not* managed by the System Coherency Interface, this parameter enables the upstream system of a port to be snooped if the upstream is not in reset and if `acchannelen_rnf` allows it.



This parameter is for debug purpose only

Do not use this parameter instead of correctly configuring CMN.

---

Type: `bool`

Default value: `false`

**disable\_CML\_port**

If true the model won't connect the CML port when there are nodes which support the CML feature (ie CXG nodes).

Type: `bool`

Default value: `false`

**enable\_logger**

Enable PVBUSLoggers for the downstream ports in the CMN model.

Type: `bool`

Default value: `false`

**enable\_rnsam\_to\_hnf\_wider\_hash**

Enable support of wider hash for the RNSAM to HNF communication.

If this variable enabled, then `bits[47:6]` from PA used in hashing function.

By default it is `[47:12]`.

Type: `bool`

Default value: `false`

### **`force_rnsam_internal`**

Force all RNSAMs to be internal independently of the mesh topology.

Type: `bool`

Default value: `true`

### **`hnf_mpam_idr_override`**

Set to override `hnf_mpam_idr[31:24]` value. Bit[28] is Reserved and is ignored.

Type: `uint64_t`

Default value: `0`

### **`mesh_config_file`**

Name of a file containing mesh placement of CMN650R2 components.

Type: `string`

Default value: `""`

### **`periphbase`**

Value for `PERIPHBASE`. Bits `[27:0]` are treated 0.

Type: `uint64_t`

Default value: `0x20000000`

### **`print_cmn_ccix_config`**

Print information about the CCIX configuration.

Type: `bool`

Default value: `false`

### **`print_cmn_config`**

Print the mesh topology and children pointers acquired from the YML file.

Type: `bool`



Default value: `false`

### **revision**

Component revision.

Currently supports r2p0.

Type: `string`

Default value: `"r2p0"`

### **rnf\_sci\_enable**

For each rnf port, indicates if the port is managed by System Coherency Interface for coherency domain entry/exit

**1**

Managed by SCI

**0**

Managed by Software

The input value is a string, for example `0xffff` or `ffff`.

Type: `string`

Default value: `"0x0"`

### **rni\_sci\_enable**

For each rni port, indicates if the port is managed by System Coherency Interface for coherency domain entry/exit

**1**

Managed by SCI

**0**

Managed by Software

The input value is a string, for example `0xffff` or `ffff`.

Type: `string`

Default value: `"0x0"`

### **show\_banner**

Show component banner:

**0**

supress entire banner

- 1
- suppress config file
- 2+
- show full banner.

Type: `uint64_t`

Default value: `2`

**skip\_cm\_n\_config\_check**

Skip any topology configuration checks. The maximum number of devices per type not verified.

Type: `bool`

Default value: `false`

**use\_yaml\_periphbase**

Use yaml param `CFGM_PERIPHBASE_PARAM` to specify periphbase address.

If false, model parameter `periphbase` will be used.

Type: `bool`

Default value: `false`

### 3.143 CMN700

Defined in `LISA/CMN700.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support
r1p0	Full support
r2p0	Full support
r3p0	Full support
r3p3	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

#### Changes in 11.31.15

The following ports were added:

- `a4s_inputs`
- `a4s_outputs`

The following ports were removed:

- `ic_dr_a4s`
- `ic_rd_a4s`

## About CMN700

- Major IP revisions (rX) are modeled and are controlled by the `revision` parameter or the topology file.
- Minor IP revisions (pY) are not modeled. There are no functional differences in the model between pY revisions, with the exception of the `periph_id2` register.
- The topology YAML 'version' field overrides the model parameter 'revision' when selecting the CMN revision.
- To configure the model, you must have installed Arm Socrates. The `mesh_config_file` parameter defines the mesh placement of the CHI nodes. Set it to the name of the yml configuration file emitted by the Socrates export process. Fast Models requires the configuration file to pass Design Rule Check (DRC) in Socrates and does not support manually editing the configuration file. You must use version SYSOC-BN-00001 r1p6-02lac1 of Socrates to generate the file. For more information about Socrates, see [Arm Socrates](#) on Arm Developer. To download it, contact Arm Technical Support.
- Interconnect models are based on the TRM description and do not typically model RTL defects.
- The maximum mesh size supported is X=12, Y=12.
- The mapping between the port number for `rnf`, `rni/rnd`, `hni`, and `snf/sbsx` interface ports and `nodeId` is based on the `nodeId` index. For example, RNF2 controls `pvbus_s_rnf[2]`. Its index is specified in the `node_info` register as `logical_id`. Similar behavior can be expected for HN-I.

If both RN-D and RN-I nodes are present, then all starting `rni` ports are mapped to RN-D nodes and then the RN-I nodes. For example, for CMN700 with two RN-D nodes, one RN-I node, and given each RN-I or RN-D node controls three interface ports, `pvbus_s_rni[0-2]` maps to `RND0`, `pvbus_s_rni[3-5]` maps to `RND1` and `pvbus_s_rni[6-8]` maps to `RNI0`.

Similarly, SN-F and SBSX nodes are mapped to `pvbus_m_snf[]` ports where starting ports are mapped to SN-F and then SBSX nodes.

- See the CMN700r3 TRM for the `HN*_SLC_SIZE_PARAM` values.
- There is limited support for RNSAMs external to the CMN. See the Limitations section for more information.
- CCG device id is CCG id + 1 when `CAL2` and `PCIE_ENABLE` are set for port 1.
- CXL Type-3 (CXL.mem) devices can be connected to the `pvbus_m_cxs[]` ports. The CXRH nodes, if any, are connected to `pvbus_m_cxs[0]` ... followed by the CCG nodes, if any.
- The Address Based Flush (ABF) feature is available. After it has started, the ABF completes instantly. The ABF implementation has the following limitations:
  - In the Fast Model, any HN-F in the interconnect can perform an ABF on any address range even if that range is not part of the System Cache Group (SCG) that the HN-F belongs to.
  - Clearing the OCM with an ABF is not supported. A trace has been added to highlight this.

- When `HNF_ABF_PR.abf_mode` is Reserved and `HNF_ABF_PR.abf_enable` is set, an error is printed and the ABF status register is set to complete successfully, but no operation takes place.
- Because an ABF completes instantly, the `SF_PM_TRANSITION_ABORT` state cannot be reached. Contact Arm Technical Support with questions about support for this error condition.
- The optional interrupt `INTREQPPU` is not supported.
- There is limited support for RAS:
  - Error logging and reporting functionality for HN-I, SBSX, XP, and MTU are supported.
  - RAS-related interrupts (`INTREQERRS`, `INTREQFAULTS`, `INTREQFAULTNS`, `INTREQERRNS`) have been added.
  - Central RAS interrupt-handling functionality of HN-D is supported.

**Note**

Enabling RAS can impact performance. To avoid this, RAS is disabled by default in the CMN700 model. To enable RAS, set the `enable_ras` parameter to true.

- There is support for A4S Multichip routing, with limitations:
  - When the `enable_a4s` parameter is false, top-level model ports are terminated with abort handlers.
  - Set `enable_a4s` parameter to true to opt into the feature.
  - The model routes A4S transactions from the `a4s_inputs` port to `tx_cxs_a4s` ports according to A4S LDIDs.
  - There is 1 A4S tx/rx port for each CCG up to a maximum of 32 CCGs.
  - When the remote transaction arrives at the receiving remote CMN on `rx_cxs_a4s` ports, it is routed to the GIC A4S port (`a4s_outputs`).
  - The A4S LDID for the CCGs can be found by reading the `CCG_RA.unit_info` register or through model parameter `print_cmn_config`.

A4S support limitations are listed in the Limitations section.

There is support for DSU AXU and DMC AXU interfaces:

- The DSU AXU and DMC AXU memory regions' start address are with the `dsu_periphbase` and `dmc_periphbase` parameters respectively.

The following list describes the level of support in the CMN700 model for different revision-specific features of the IP:

- r1p0, second release

### **CXLv2.0 host-side support for CXL.mem and CXL.io protocols for Type-3 memory expansion devices**

Model supports Type-3 connections using PVBUS

**32 CCG or CXG gateway nodes**

Supported

**Non power-of-2 hashing of HN-Fs with  $2N * \{1, 3, \text{or } 5\}$  up to 64 HN-Fs or 128 HN-Fs with CAL**

Supported

- r2p0, third release

**Remote PCIe streaming support**

Not in scope

**1.5MB SLC support**

Supported, SLC\_SIZE=2, NUM\_WAYS=12

**90 RN-I support**

Only 40 supported (3 AXI port each)

**128 SN-F/SBSX support**

Only 80 supported

**AXID based for port aggregation across chip**

Not supported

**RNSAM support for 4 chip flat hashing configuration**

Supported

- r3p0, fourth release

**AXU port on MXPs**

Not supported

**512 RN-I requests support**

Not in scope

**16-bit REQ RSVDC support**

Width reported in info\_global register; RSVDC not in scope

**Configurable write cancel threshold in RN-I and RN-D**

Not in scope

**Remote DVM sync collapsing**

Not in scope

**CPAG MOD-3 hashing**

CPAG hashing not supported

**PCIe write streaming improvements**

Not in scope

- r3p3, fifth release

**Performance optimization guideline improvements for RN-I and RN-D**

Not in scope

**HN-P and HN-I AxID Encoding improvements**

Not in scope

## HCAL2 Discovery supported

Feature available, but not tested for cmn700

### Limitations

- Out of scope:
  - PMU counters are not supported. Counter registers are implemented as **RAZ**.
  - QoS is not supported and all related registers are **RAZ/WI**.
  - Error injection and error generation are not supported. All error registers are **RAZ/WI**.
  - Power, clock, and interrupt signals are not supported, but RAS-related interrupts are supported.

The P-channel (power) signals PREQ\_LOGIC, PSTATE\_LOGIC, PACCEPT\_LOGIC, PDENY\_LOGIC, and PACTIVE\_LOGIC, are not implemented. The model behaves as if PSTATE\_LOGIC is 5'b00000. So the initial power state of the HN-F nodes is NOSFSLC/OFF. This is reflected in the reset values of `cmn_hns_ppu_pwpr`, which was named `por_hnf_ppu_pwpr` in earlier IP revisions. Software must write the `*_ppu_pwpr` registers to change the power state. Both HAM and FAM power states enable all of the SLC ways.

- Protocol credits and flit buffers are not supported.
- Snoop filtering is not supported.
- Prefetch target operations are not supported.
- Non-XY routing behavior is not supported.
- The MTE feature has been minimally tested and the functionality cannot be guaranteed. MTE error injection and detection are not supported.
- An access to a reserved or nonexistent register does not abort. It returns `pv::Tx_Data::TX_OK` and is **RAZ/WI**.
- Topologies with three I/O masters connected through an RN-I bridge device have been tested. Configurations where multiple I/O masters share a single ACE-lite port have not been tested.
- RNSAMs external to the mesh network are not functionally supported.



Note

Model parameter `force_rnsam_internal` is true by default. If this parameter is set to false and the topology has an RNFx (non-ESAM), the external bit (bit[31]) of the RNSAM children pointer is enabled. This bit is the only functionality governed by this parameter.

- By default, HN-F hashing uses the address [MAX:12] instead of the actual address [MAX:6], due to the DMI mechanism in the model. Enable the parameter `enable_rnsam_to_hnf_wider_hash` to make hashing logic use the actual address [MAX:6], but this might reduce the simulation speed of the model.
- OKB SLC data RAM and tag RAM are not supported.
- Source-based SLC cache partitioning is not supported.
- Way-based SLC cache partitioning is not supported.

- SLC/LCC partitioning dynamic allocation is not supported.
- Remote chip addresses incorrectly allocate a line in the SLC when `cache_state_modelled=true`. This can cause unexpected cache behavior that does not match the RTL, exhausting the SLC earlier or extra evictions.
- The fields in the `mxp_device_port_connect_info` register corresponding to the number of device credit slices and the number of CAL credit slices are not supported.
- HN-Fs with different SLC sizes in the same configuration are not supported.
- GIC communication over A4S ports is not supported.
- No support or updates for the following parameters:
  - `POR_RSVDC_STRONGNC_EN_PARAM`
  - `POR_HNSAM_CUSTOM_REGS_PARAM`
- No updates for a new bit in `CMN_HNS_CFG_CTL` to disable HNS stashing snoop (`hnf_stash_snp_dis`).
- HND-APB registers not supported.
- HN-P nodes are not supported as hashed target from the RNSAM.
- There is limited support for CXL Type-3. It only supports a single device connection (`sa_ports_cnt`).
- For CMN700R1, `por_hnf_cfg_ctl` follows the CMN700R0 write mask and reset value.
- For CMN700R1 and later, stash snooping is not supported.
- The model cannot activate both CCG APB register access traces and CMN register access traces simultaneously. Use the parameter `register_traces_for_ccg_apb_accesses` to enable CCG APB register access traces. By default, CMN register access traces are available for activation.
- MPAM features are not supported in the model, and the configuration-dependent register reset values do not match the RTL. Some of the registers (`por_hnf_mpam_idr` to `por_hnf_mpam_mbwumon_idr`) shared between security modes are not present in MPAM\_S.
- The MPAM reset values described in `por_hnf_mpam_ns_por_hnf_mpam_idr[31:24]` default to the not-supported state. The model parameter `hnf_mpam_idr_override` can be used to override this but does not implement any of the functionality.
- The following limitations apply to System Cache Groups and Hash Target Groups:
  - The `hashed_target_grp_hnp_nodeid_reg` which is used for CCG/HN-P target IDs is not supported.
  - AXID hashing across HN-P/CCGs is not supported.
- `MPAM_S_secure_register_groups_override` is not supported.
- No specific Debug Trace (DT) node functionality is supported. See [Plug-ins for Fast Models](#) for information about using plug-ins for trace.
- For Power State Transition related flush/disable of the SLC due to `HNF_PPU_PWPR` register, only the Operational Mode field (bits[7:4]) is considered. The Power Policy field (bits[3:0]) is ignored.

- Setting HAM mode for the Operational Mode field (bits[7:4]) of the HNF\_PPU\_PWPR is equivalent to FAM mode. The SLCH2 is neither flushed nor disabled according to the Power Policy field (bits[3:0]).
- Only hashing granularities of 4096B and above are supported.
- All RNs are assumed to have the same HN mapping. Programming RNSAMs differently between RNs is not supported.
- RNSAM cannot be programmed with the same HN-F target in a hashed and non-hashed memory region.
- `RNSAM_STATUS.default_target_type` and `RNSAM_STATUS.default_nodeid` are used as the default target only when `RNSAM_STATUS.use_default_node=1`. Otherwise, when a txn is sent into the model that does not belong to any of the address regions programmed in the RNSAM, it is routed to the HN-D irrespective of what is programmed in the `RNSAM_STATUS.default_target_type` and `RNSAM_STATUS.default_nodeid` fields.
- Address bit masking is not supported in RNSAM or HN-F SAM.
- The OCM does not keep track of the total memory used across the S/NS security world and does not perform any eviction either. It is up to the software to make sure it does not oversubscribe the OCM.
- RSVDC StrongNC and its associated functionality is not supported.
- User-defined hashing mechanism in an SCG is not supported.
- The CXSA mode has limited support. Currently, it only supports one aggregated device.
- The model does not display any register traces.
- GenericTrace for the CMN700 Fast Model, incorrectly mentions “CMN600” in logs.
- CCG node addresses do not match RTL node addresses if not using node addresses from yml.
- The following limitations are specific to revision r2p0:
  - The model does not support the RA\_PRESENT configurable option. RAs are always present in CCG.
  - Maximum number of RN-D supported is 40.
- The following limitations apply to revisions r2p0 and r3p0:
  - No support for RWL (ReadWriteLock).
  - Maximum number of RN-I supported is 40.
  - Maximum combined number of RN-I and RN-D is 40.
  - SN-Fs on CAL4 are not supported.
  - Maximum number of SN interfaces supported is 80.
- The following limitations are specific to revision r3p0:
  - Maximum RAID of 1024 is not supported.
  - Direct Subordinate Access (DSA) CCG inbound request bypass of HN-F is not supported.
  - CXL v2.0 device support for various types is not supported or verified.
  - CXL v2.0 host support for various types is not supported or verified.



- There is no support for CPAG MOD-3 hashing.
- AXU Limitations:
  - There is no support for AXU on MXPs.
  - There is no support for APB accesses on AXU interfaces.
  - The `dsu_apb_only` and `dmc_axu_only` fields are not supported.
- RAS feature limitations:
  - Error logging and reporting functionality for CCG, HN-F, and CXHA are not supported.
  - Single-bit error injection for MTU is not supported as there is no ECC checker or register present to support it.
  - NDE response and Poison error check are not supported.
  - Flit parity and Data check errors are not supported.
  - The information that is captured as source ID, target ID, and logical ID in the ERRMISC register might not be correct or match the hardware.
  - HN-D Illegal Configuration check does not check that the access is of device type.
  - HN-D Illegal Configuration check does not check the access security mode.
  - SN-F RAS errors are treated as SBSX errors.
- A4S support limitations:
  - GIC\_DESTID input strap is not supported. Incoming transactions from remote chip are always routed to IC\_RD.
  - The model assumes the presence of 1x A4S port for GIC without regard for the actual number of a4s interfaces in `themesh_config_file` topology.
  - The model does not require user software at runtime to enable the CMN to route multichip A4S transactions between chips.
  - Limited performance testing has been performed.
  - There is no support for the use of “id\_map” file specified by CMN Configuration Integration Manual (CIM) to configure the model for reset.
- Model behavior does not reflect errata notice 2732981. The model behaves as r3p1. See the errata for details.
- For an HTG/SCG, the model does not consider the secondary region if the primary region is not valid.
- HN-D is only permitted on device port P2 in a single-MXP configuration.
- Only the last RN-F in the mesh, which is the one with the highest logical id, can be controlled by `mvp_p[0-5]_syscoreq_ctl` registers. Also, incorrectly, it can be controlled from any XP.
- HNSAM only supports two non-hashed memory regions. Memory regions programmed using `cmn_hns_sam_nonhash_cfg[1|2]_memregion2-63` registers are ignored.
- The Hierarchical hashing fields  
`HASHED_TARGET_GRP_HASH_CNTL_REG.htg_region#{index}_hier_enable_address_stripping`  
 and `HASHED_TARGET_GRP_HASH_CNTL_REG.htg_region#{index}_hier_cluster_mask` are not supported.

- The `HNSAM_DEF_HASHED_GRP_EN` yml parameter is not supported and HN-F SAM legacy mode is always enabled.
- The `HNSAM_NUM_HTG` yml parameter is not supported.
- When both `POR_CCLA_ULL_CTL.u11_to_u11_en` and `POR_CCLA_ULL_CTL.send_vd_init` bits are set then both `POR_CCLA_ULL_STATUS.tx_state` and `POR_CCLA_ULL_STATUS.rx_state` are set. The other side of the link is not consulted to set `POR_CCLA_ULL_STATUS.rx_state`.
- HN-F SAM:
  - Address masking in default hash regions in HN-F SAM is not supported.
  - Hashing in default hash regions in HN-F SAM is not supported. However, trace `SNF_HASHING_Target_SNF` displays the target SN-F as if 3SN/6SN hashing were enabled.



The transaction is always routed to the SN0 nodeid programmed in the `SAM_CONTROL` register.

- Re-programming regions in HN-F SAM is not tested.
- Hashing across CCGs in HN-F SAM is not supported.
- Address compare hashed regions in HN-F SAM do not support non-power of 2 hashing. When non-power of 2 hashing is enabled, the first SN in the HTG (SN0) is used as the target.
- `HNSAM_DEF_HASHED_GRP_EN` yml parameter is not supported.
- CMN700 r1p0 supports only 8 hashed regions in HN-F SAM. CMN700 r2p0 and r3p0 support 16 hashed regions.
- Default hash regions in HN-F SAM have limited support.  
`cmn_hns_sam_cfg1_def_hashed_region` and `cmn_hns_sam_cfg2_def_hashed_region` are not supported.
- Hashing across SN-F on CAL2 in HN-F SAM is not supported.
- Remote chip addresses are incorrectly allocating a line in the SLC when `cache_state_modelled=true`. This can cause unexpected cache behavior that does not match the RTL, exhausting the SLC earlier or extra evictions.
- Model does not support APB registers.
- `POR_MTU_TAG_ADDR_CTL.memory_map_mode=3'b000` (Pass-through) is the only supported behavior. Other values for this field are not supported.
- For RN-D nodes, when software writes `SYSCOREQ`, DVM propagation gets enabled but `SYSCOACK` is not set.
- Functionality behind 'dn\_domain' yml param is abstracted away and the model forwards the DVM message to all upstream ports irrespective of 'dn\_domain' value. Reading `rnsam_status.dn_nodeid` does not reflect what `dn_domain` the node is in.

### About the `debug_force_snoop` parameter

The CMN interconnect normally starts up with snooping disabled.

The parameters `rnf_sci_enable` and `rni_sci_enable` determine which connections are managed by the System Coherency Interface. Connections that are managed by the System Coherency Interface look after themselves for entering and leaving the coherency domain and `debug_force_snoop` has no effect on those ports.

However, for connections that are not managed by the System Coherency Interface, `debug_force_snoop` enables the system upstream of a port to be snooped if the upstream is not in reset and if `acchannelen_rnf` allows it.

Therefore, software does not need to know how to turn snoops in the interconnect on and off. The reset state of the upstream rnf ports is determined by the `rnf_upstream_reset_state[]` signals and must be connected for any port that could go into reset and you want managed by `debug_force_snoop`.

If the connection is managed by SCI or can never go into reset then you need not connect this and the interconnect assumes that the upstream system is not in reset and will always enable the snoops if `ACCHANNELENSx` allows it.

If you fail to connect `rnf_upstream_reset_state[]` correctly, then the upstream system might receive snoop messages while in reset and it will complain that it received a snoop request while it was in reset.

If software overrides the enables then the next change to `rnf_upstream_reset_state[]` or the interconnect reset will overwrite the software values.

Using this operation along with software controlling the enables could confuse the software. This option does not remove the responsibility of the upstream system to clean any shared dirty data from its caches before going into reset.



Note

This parameter is for debug purposes only. Do not use it instead of correctly configuring CMN.

## Iris and MTI instances for CMN700

This model has the following Iris instances:

Name	Instance type
CMN700	CMN700
CMN700.bus_slave_ocm_NS	PVBusSlave
CMN700.bus_slave_ocm_S	PVBusSlave
CMN700.cmn600_cache	PVCache
CMN700.cmn600_cache.downstream[Z].pvbusmaster (where Z = 0-13)	PVBusMaster
CMN700.cmn600_cache.upstream[Z] (where Z = 0-37)	PVBusSlave
CMN700.cmn700_tag_cache	CMN_TAG_CACHE
CMN700.cmn700_tag_cache.metadata_controllerZ (where Z = 0-127)	PVMetadataController

Name	Instance type
CMN700.cmn700_tag_cache.metadata_controllerZ.MetaDataMapper (where Z = 0-127)	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controllerZ.mdc_downstream_port[0].pvbusmaster (where Z = 0-127)	PVBusMaster
CMN700.cmn700_tag_cache.remapperZ (where Z = 0-127)	PVBusMapper
CMN700.dmc_axu_mapper	PVBusMapper
CMN700.dsu_axu_mapper	PVBusMapper
CMN700.hnf_exclusive_monitor	PVBusExclusiveMonitor
CMN700.hnf_exclusive_monitor.bus_mapper	PVBusMapper
CMN700.hni_exclusive_monitorY (where Y = 0-3)	PVBusExclusiveMonitor
CMN700.hni_exclusive_monitorY.bus_mapper (where Y = 0-3)	PVBusMapper
CMN700.ocm_decoder	PVBusMapper
CMN700.ocm_exclusive_monitor	PVBusExclusiveMonitor
CMN700.ocm_exclusive_monitor.bus_mapper	PVBusMapper
CMN700.snf_mapper	PVBusMapper

This model has the following MTI trace components:

Name	Component type
CMN700	CMN700
CMN700.bus_slave_ocm_NS	PVBusSlave
CMN700.bus_slave_ocm_S	PVBusSlave
CMN700.cmn600_cache	CMN600Cache
CMN700.cmn600_cache.downstream[Z].pvbusmaster (where Z = 0-13)	PVBusMaster
CMN700.cmn600_cache.upstream[Z] (where Z = 0-37)	PVBusSlave
CMN700.cmn700_tag_cache	CMNTAGCACHECADI
CMN700.cmn700_tag_cache.metadata_controllerZ (where Z = 0-127)	MetaDataController
CMN700.cmn700_tag_cache.metadata_controllerZ.MetaDataMapper (where Z = 0-127)	PVBusMapper
CMN700.cmn700_tag_cache.metadata_controllerZ.mdc_downstream_port[0].pvbusmaster (where Z = 0-127)	PVBusMaster
CMN700.cmn700_tag_cache.remapperZ (where Z = 0-127)	PVBusMapper
CMN700.dmc_axu_mapper	PVBusMapper
CMN700.dsu_axu_mapper	PVBusMapper
CMN700.hnf_exclusive_monitor	PVBusExclusiveMonitor
CMN700.hnf_exclusive_monitor.bus_mapper	PVBusMapper
CMN700.hni_exclusive_monitorY (where Y = 0-3)	PVBusExclusiveMonitor
CMN700.hni_exclusive_monitorY.bus_mapper (where Y = 0-3)	PVBusMapper
CMN700.ocm_decoder	PVBusMapper
CMN700.ocm_exclusive_monitor	PVBusExclusiveMonitor
CMN700.ocm_exclusive_monitor.bus_mapper	PVBusMapper
CMN700.snf_mapper	PVBusMapper

## Ports for CMN700

Port	Direction	Protocol	Description
a4s_inputs	slave	PVBus	AXI4Stream input ports.
a4s_outputs	master	PVBus	AXI4Stream output ports.
event_clusters	peer	Signal	CPU event communication signal from the clusters.
event_downstream_link_signal	master	Signal	CPU event communication signal to the CMLHub. Event from the CMN towards the Hub NOTE that these are virtual “links” on underlying PCIe hardware bus.
event_upstream_link_signal	slave	Signal	Event from the Hub towards the CMN
intreqerrns_irq_out	master	Signal	Interrupt signal
intreqerrs_irq_out	master	Signal	Interrupt signal
intreqfaultns_irq_out	master	Signal	Interrupt signal
intreqfaults_irq_out	master	Signal	Interrupt signal
pvbus_m_cml_cfg	master	PVBus	CML downstream hub configuration port
pvbus_m_cml	master	PVBus	CML downstream ports
pvbus_m_cxs	master	PVBus	CXS downstream ports
pvbus_m_dmc_axu	master	PVBus	DMC AXU ports
pvbus_m_dsus_axu	master	PVBus	DSU AXU ports
pvbus_m_hni	master	PVBus	HNI downstream ports.
pvbus_m_snf	master	PVBus	SNF downstream port.
pvbus_s_apb	slave	PVBus	APB interface port.
pvbus_s_ccg_apb	slave	PVBus	CCG APB interface port.
pvbus_s_cml	slave	PVBus	CML upstream ports
pvbus_s_rnf	slave	PVBus	RNF upstream ports.
pvbus_s_rni	slave	PVBus	RNI upstream ports. NOTE the upper 150 ports are only used in r2/r3.
reset_in	slave	Signal	Reset signal.
rnf_sci_s	slave	SystemCoherencyInterface	System Coherency port to move the RN-F in and out of the coherency domain.
rnf_upstream_reset_in	slave	Signal	Used by the interconnect to determine the reset state of the RNF manager to check an aspect of reset sequencing. If debug_force_snoop is used, this port must be connected.
rni_sci_s	slave	SystemCoherencyInterface	System Coherency port to move the RN-D in and out of the coherency domain.
rni_upstream_reset_in	slave	Signal	Used by the interconnect to determine the reset state of the RNI manager to check an aspect of reset sequencing. If debug_force_snoop is used, this port must be connected. NOTE the upper 150 ports are only used in r2/r3.
rx_cxs_a4s	slave	PVBus	Receive channel of A4S packets from a remote CMN.
tx_cxs_a4s	master	PVBus	Transmit channel of A4S packets to a remote CMN.

## Parameters for CMN700

### **a4s\_logicalid**

A4S ID mapping of the GIC destination component connected through a CCG port.

Specify the `CCG_NODE_ID` and the destination A4S Logical ID of the GIC component connected by using a decimal number format like:

```
<CCG_NODEID0>=<A4S_LID0>,<CCG_NODEID1>=<A4S_LID1>
```

For example for CCG Node ID 54 with A4S ID 12 - 54=12.

All of the CCG nodes must be specified.

The parameter is only valid when the `enable_a4s` is also enabled. The default behavior without this parameter is to automatically assign an incrementing A4S ID.

Type: `string`

Default value: ""

### **acchannelen\_rnf**



DEPRECATED: Will be removed after FM 11.18. Use `rnf_sci_enable` instead.

For each `rnf` port, indicates if the port is populated with a snoop responding device or not.

The input value is a string, for example `0xffff` or `ffff`.

Type: `string`

Default value: "0"

### **acchannelen\_rni**



DEPRECATED: Will be removed after FM 11.18. Use `rni_sci_enable` instead.

For each `rni` port, indicates if the port is populated with a dvm responding device or not.

The input value is a string, for example `0xffff` or `ffff`.

Type: `string`

Default value: `"0"`

### **cache\_state\_modelled**

Model the cache state.

Type: `bool`

Default value: `false`

### **debug\_force\_snoop**

The CMN700 interconnect will normally start with snooping disabled.

The parameter `rnf_sci_enable` and `rni_sci_enable` determines which connections are managed by the System Coherency Interface.

For connections that are managed by the System Coherency Interface, then they will look after themselves for entering and leaving the coherency domain and this parameter has no effect for those ports.

However, for connections that are *not* managed by the System Coherency Interface, this parameter enables the upstream system of a port to be snooped if the upstream is not in reset and if `acchannelen_rnf` allows it.



This parameter is for debug purpose only

Do not use this parameter instead of correctly configuring CMN.

---

Type: `bool`

Default value: `false`

### **disable\_CML\_port**

If true the model won't connect the CML port when there are nodes which support the CML feature (ie CXG nodes).

Type: `bool`

Default value: `false`

### **dmc\_periphbase**

Value for DMC\_PERIPHBASE.

Type: `uint64_t`

Default value: `0xffffffffffffffff`

### **`dsu_periphbase`**

Value for DSU\_PERIPHBASE.

Type: `uint64_t`

Default value: `0xffffffffffffffff`

### **`enable_a4s`**

Enables A4S ports for GIC multi-chip routing.

Type: `bool`

Default value: `false`

### **`enable_logger`**

Enable PVBUSLoggers for the downstream ports in the CMN model.

Type: `bool`

Default value: `false`

### **`enable_ras`**

Enables RAS. There is an impact on performance when RAS is enabled.

Type: `bool`

Default value: `false`

### **`enable_rnsam_to_hnf_wider_hash`**

Enable support of wider hash for the RNSAM to HNF communication.

If this variable enabled, then `bits[47:6]` from PA used in hashing function.

By default it is `[47:12]`.

Type: `bool`

Default value: `false`

### **`force_rnsam_internal`**

Force all RNSAMs to be internal independently of the mesh topology.

Type: `bool`

Default value: `true`



**hnf\_mpam\_idr\_override**

Set to override `hnf_mpam_idr[31:24]` value. Bit[28] is Reserved and is ignored.

Type: `uint64_t`

Default value: 0

**mesh\_config\_file**

Name of a file containing mesh placement of CMN700 components.

Type: `string`

Default value: ""

**periphbase**

Value for `PERIPHBASE`. Bits [27:0] are treated 0.

Type: `uint64_t`

Default value: `0x20000000`

**print\_cmn\_ccix\_config**

Print information about the CCIX configuration.

Type: `bool`

Default value: `false`

**print\_cmn\_config**

Print the mesh topology and children pointers acquired from the YML file.

Type: `bool`

Default value: `false`

**register\_traces\_for\_ccg\_apb\_accesses**

Will be removed when enhancement SDDKW-74284 is done.

---

Intended for use with trace plugins.

**true**

registers traces to CCG register accesses through CCG APB interface.

**false**

registers traces to CMN register accesses through all other interfaces (eg RN nodes).

Type: `bool`

Default value: `false`

**revision**

Component revision.

Currently supports r0p0, r1p0, r2p0, r3p0, r3p3.

Type: `string`

Default value: `"r0p0"`

**rnf\_sci\_enable**

For each rnf port, indicates if the port is managed by System Coherency Interface for coherency domain entry/exit

**1**

Managed by SCI

**0**

Managed by Software

The input value is a string, for example `0xffff` or `ffff`.

Type: `string`

Default value: `"0x0"`

**rni\_sci\_enable**

For each rni port, indicates if the port is managed by System Coherency Interface for coherency domain entry/exit

**1**

Managed by SCI

**0**

Managed by Software

The input value is a string, for example `0xffff` or `ffff`.

Type: `string`

Default value: `"0x0"`

**show\_banner**

Show component banner:

- 0
- supress entire banner
- 1
- suppress config file
- 2+
- show full banner.

Type: `uint64_t`

Default value: `2`

**skip\_cmn\_config\_check**

Skip any topology configuration checks. The maximum number of devices per type not verified.

Type: `bool`

Default value: `false`

**use\_yaml\_periphbase**

Use yaml param `CFGM_PERIPHBASE_PARAM` to specify periphbase address.

If false, model parameter `periphbase` will be used.

Type: `bool`

Default value: `false`

**yaml\_has\_node\_addresses**

Does the top-level YML file describe node-addresses ?.

Type: `bool`

Default value: `false`

### 3.144 CMN\_S3

Defined in `LISA/CMN_S3.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0	Preliminary support
r1	Preliminary support
r2	Preliminary support

For an explanation of the quality levels, see [Quality level definitions](#).

## Changes in 11.31.15

The following ports were added:

- `a4s_inputs`
- `a4s_outputs`

The following ports were removed:

- `ic_dr_a4s`
- `ic_rd_a4s`

## About CMN\_S3

- Major IP revisions (rX) are modeled and are controlled by the `version` parameter in the topology file.
- Minor IP revisions (pY) are not modeled. There are no functional differences in the model between pY revisions, with the exception of the `periph_id2` register.
- The topology YAML 'version' field overrides the model parameter 'revision' when selecting the CMN revision.
- To configure the model, you must have installed Arm Socrates. The `mesh_config_file` parameter defines the mesh placement of CHI nodes. Set it to the name of the yml configuration file emitted by the Socrates export process. Fast Models requires the configuration file to pass Design Rule Check (DRC) in Socrates and does not support manually editing the configuration file. You must use r1p7-06 version of Socrates to generate the file. For more information about Socrates, see [Arm Socrates](#) on Arm Developer. To download it, contact Arm Technical Support.
- Interconnect models are based on the TRM description and do not typically model RTL defects.

The following functionality is supported:

- CHI Issue F Realm Management Extension (RME) with `LEGACY_TZ_EN`.
  - `LEGACY_TZ_EN` is controlled by a model parameter.
    - RCR registers are **RAZ/WI**.
    - Does not prevent REALM/ROOT transactions from flowing through interconnect.
- HNS device isolation.
- Hybrid CAL3 for discovery only.
- CPA groups including multi CPA groups.
- Large systems can be built by connecting multiple CMNs (Coherent Mesh Networks) using their CCG components. These are referred to as “multi-chip” systems. In real hardware the CCG components communicate using the CXS protocol. The CMN Fast Models do not implement CXS, but PVBUS can be used to represent CCG connections.
  - `pvbuss_m_cxs` and `pvbuss_s_cxs` ports represent the CCG ports

- Connecting `cmnA.pvbus_m_cxs[a]` to `cmnB.pvbus_s_cxs[b]`, represents connecting `cmnA.CCGa` to `cmnB.CCGb`; the reverse connection should also be made: `cmnB.pvbus_m_cxs[b]` to `cmnA.pvbus_s_cxs[a]`.
- Caching functionality (and consequently snooping) is not supported in multichip platforms.
- DVM's work correctly only in systems where all CMN's are connected to all other CMN's in the system.
- `pvbus_m_cxs` ports can also be used to connect to CXL Type-3 devices.
- No support for the CMLHub to model multi-chip systems.
- MXP AXU interfaces are supported.
  - The base address for accesses on these interfaces must be specified through the `mvp_axu_periphbase` parameter.
- RAS Pseudo-Fault Generation (PFG) for HNS and HNI nodes.
- System RAS Agents when RAS 2.0 mode is enabled.

## Limitations



Issues listed in this section have identifiers of the form [SDDKW-x]. These are Arm internal references only.

- [SDDKW-82504] [CMN700](#) limitations apply to CMN\_S3 model as well, in addition to those listed below. See [CMN700](#) for limitations.
- RAS PFG functionality is not supported on all node types.
- CFGM RAS behavior only present on main HND node.
- Features not supported:
  - CHI C2C is not supported.
  - Direct Subordinate Access (DSA)
  - MTSX
  - Port to Port Forwarding for CML SMP
  - Memory Protection Engine (MPE) for CXL-Type Host
- Out of scope features:
  - Datasource handling
- [SDDKW-81128] HNS device isolation:
  - Hashed target groups are checked for disabled nodes at the time the RNSAM is programmed and ready (`rnsam_status.install_req=1'b1`). Disabling HNS devices after RNSAM programming is a no-op.
  - Having disabled HNS nodes in a hash target group is a non-fatal error. Model displays an error message and behaves the same as if the node was enabled.

- The default scenario is not supported, i.e. disabling only one device of a CAL2 pair in a hash target group triggers a non-fatal error message for that specific device of the CAL2 pair. The error message does not affect the other device behind CAL2.
- [SDDKW-82529] OCM does not support RL/RT PAS
- [SDDKW-76061] Model does not support global secure overrides for root registers.
- [SDDKW-76516] Model does not support RCR (root override register) for the following nodes:
  - DT
  - DN
  - MTU
  - APB
- [SDDKW-80175] DSU HNI region in RNSAM is not supported.
- [SDDKW-79842] Increased maximum HN-I limit to 48. Maximum tested is 46.
- [SDDKW-85869] `RNSAM_STATUS.default_target_type` and `RNSAM_STATUS.default_nodeid` is used as the default target only when `RNSAM_STATUS.use_default_node=1`. Otherwise, when a `txn` is sent into the model which does not belong to any of the address regions programmed in the RNSAM, it is routed to the HND irrespective of what is programmed in the `RNSAM_STATUS.default_target_type` and `RNSAM_STATUS.default_nodeid` fields
- [SDDKW-85945] `HASHED_TARGET_GRP_HASH_CNTL_REG0-31.htg_region#{index}_hier_enable_address_striping` and `HASHED_TARGET_GRP_HASH_CNTL_REG0-31.htg_region#{index}_hier_cluster_mask` are not supported.
- Hybrid CAL
  - [SDDKW-90932] CAL2 has not been tested.
  - [SDDKW-90932] CAL3 functionality beyond discovery has not been tested.
- [SDDKW-85839] Register bitfields with W1S are treated as RW.
- HNP on CAL4 are not supported.
- HNS on CAL4 are not tested.
- [SDDKW-91300] HCAL3 is not supported.
- `NUM_EXCL_CHIPS` and `MAX_EXCL_ALL_CHIPS` parameter values are not reflected in register `por_info_global_1`.
- `LCNSAM_NUM_HTG`, `LCNSAM_NUM_NONHASHGROUP` parameter values are not reflected in register `cmn_hns_unit_info_1`.
- `NUM_VMF` parameter value is not reflected in `por_dn_build_info`.
- AXU
  - APB accesses on MXP AXU interfaces are not supported.
  - The `mxp_axu_only` field is not supported.

### Iris and MTI instances for CMN\_S3

This model has the following Iris instances:

Name	Instance type
CMN_S3	CMN700
CMN_S3.bus_slave_ocm_NS	PVBusSlave
CMN_S3.bus_slave_ocm_S	PVBusSlave
CMN_S3.cmn600_cache	PVCache
CMN_S3.cmn600_cache.downstream[Z].pvbusmaster (where Z = 0-43)	PVBusMaster
CMN_S3.cmn600_cache.upstream[Z] (where Z = 0-116)	PVBusSlave
CMN_S3.cmn_s3_tag_cache	CMN_TAG_CACHE
CMN_S3.cmn_s3_tag_cache.metadata_controllerZ (where Z = 0-127)	PVMetadataController
CMN_S3.cmn_s3_tag_cache.metadata_controllerZ.MetaDataMapper (where Z = 0-127)	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controllerZ.mdc_downstream_port[0].pvbusmaster (where Z = 0-127)	PVBusMaster
CMN_S3.cmn_s3_tag_cache.remapperZ (where Z = 0-127)	PVBusMapper
CMN_S3.dmc_axu_mapper	PVBusMapper
CMN_S3.hnf_exclusive_monitor	PVBusExclusiveMonitor
CMN_S3.hnf_exclusive_monitor.bus_mapper	PVBusMapper
CMN_S3.hni_exclusive_monitorY (where Y = 0-26)	PVBusExclusiveMonitor
CMN_S3.hni_exclusive_monitorY.bus_mapper (where Y = 0-26)	PVBusMapper
CMN_S3.ocm_decoder	PVBusMapper
CMN_S3.ocm_exclusive_monitor	PVBusExclusiveMonitor
CMN_S3.ocm_exclusive_monitor.bus_mapper	PVBusMapper
CMN_S3.snf_mapper	PVBusMapper

This model has the following MTI trace components:

Name	Component type
CMN_S3	CMN_S3
CMN_S3.bus_slave_ocm_NS	PVBusSlave
CMN_S3.bus_slave_ocm_S	PVBusSlave
CMN_S3.cmn600_cache	CMN600Cache
CMN_S3.cmn600_cache.downstream[Z].pvbusmaster (where Z = 0-43)	PVBusMaster
CMN_S3.cmn600_cache.upstream[Z] (where Z = 0-116)	PVBusSlave
CMN_S3.cmn_s3_tag_cache	CMNTAGCACHECADI
CMN_S3.cmn_s3_tag_cache.metadata_controllerZ (where Z = 0-127)	MetaDataController
CMN_S3.cmn_s3_tag_cache.metadata_controllerZ.MetaDataMapper (where Z = 0-127)	PVBusMapper
CMN_S3.cmn_s3_tag_cache.metadata_controllerZ.mdc_downstream_port[0].pvbusmaster (where Z = 0-127)	PVBusMaster
CMN_S3.cmn_s3_tag_cache.remapperZ (where Z = 0-127)	PVBusMapper
CMN_S3.dmc_axu_mapper	PVBusMapper
CMN_S3.hnf_exclusive_monitor	PVBusExclusiveMonitor
CMN_S3.hnf_exclusive_monitor.bus_mapper	PVBusMapper
CMN_S3.hni_exclusive_monitorY (where Y = 0-26)	PVBusExclusiveMonitor
CMN_S3.hni_exclusive_monitorY.bus_mapper (where Y = 0-26)	PVBusMapper

Name	Component type
CMN_S3.ocm_decoder	PVBusMapper
CMN_S3.ocm_exclusive_monitor	PVBusExclusiveMonitor
CMN_S3.ocm_exclusive_monitor.bus_mapper	PVBusMapper
CMN_S3.snf_mapper	PVBusMapper

## Ports for CMN\_S3

Port	Direction	Protocol	Description
a4s_inputs	slave	PVBus	AXI4Stream input ports.
a4s_outputs	master	PVBus	AXI4Stream output ports.
event_clusters	peer	Signal	CPU event communication signal from the clusters.
event_downstream_link_signal	master	Signal	CPU event communication signal to the CMLHub. Event from the CMN towards the Hub NOTE that these are virtual "links" on underlying PCIe hardware bus.
event_upstream_link_signal	slave	Signal	Event from the Hub towards the CMN
intreqerrns_irq_out	master	Signal	Interrupt signal
intreqerrs_irq_out	master	Signal	Interrupt signal
intreqfaultns_irq_out	master	Signal	Interrupt signal
intreqfaults_irq_out	master	Signal	Interrupt signal
pvbus_m_cxs	master	PVBus	CXS downstream ports
pvbus_m_dmc_axu	master	PVBus	DMC AXU ports
pvbus_m_dsus_axu	master	PVBus	DSU AXU ports
pvbus_m_hni	master	PVBus	HNI downstream ports.
pvbus_m_mxp_axu	master	PVBus	MXP AXU ports
pvbus_m_snf	master	PVBus	SNF downstream port.
pvbus_s_apb	slave	PVBus	APB interface port.
pvbus_s_ccg_apb	slave	PVBus	CCG APB interface port.
pvbus_s_cxs	slave	PVBus	CXS upstream ports
pvbus_s_rnf	slave	PVBus	RNF upstream ports.
pvbus_s_rni	slave	PVBus	RNI upstream ports.
reset_in	slave	Signal	Reset signal.
rnf_sci_s	slave	SystemCoherencyInterface	System Coherency port to move the RN-F in and out of the coherency domain.
rnf_upstream_reset_in	slave	Signal	Used by the interconnect to determine the reset state of the RNF manager to check an aspect of reset sequencing. If debug_force_snoop is used, this port must be connected.
rni_sci_s	slave	SystemCoherencyInterface	System Coherency port to move the RN-D in and out of the coherency domain.
rni_upstream_reset_in	slave	Signal	Used by the interconnect to determine the reset state of the RNI manager to check an aspect of reset sequencing. If debug_force_snoop is used, this port must be connected.
rx_cxs_a4s	slave	PVBus	Receive channel of A4S packets from a remote CMN.



Port	Direction	Protocol	Description
tx_cxs_a4s	master	PVBus	Transmit channel of A4S packets to a remote CMN.

Parameters for CMN\_S3

a4s\_logicalid

A4S ID mapping of the GIC destination component connected through a CCG port.

Specify the ccg\_node\_id and the destination A4S Logical ID of the GIC component connected by using a decimal number format like:

```
<CCG_NODEID0>=<A4S_LID0>,<CCG_NODEID1>=<A4S_LID1>
```

For example for CCG Node ID 54 with A4S ID 12 - 54=12.


All of the CCG nodes must be specified.

The parameter is only valid when the enable\_a4s is also enabled. The default behavior without this parameter is to automatically assign an incrementing A4S ID.

Type: string

Default value: ""

acchannelen\_rnf



Note

DEPRECATED: Will be removed after FM 11.18. Use rnf\_sci\_enable instead.


For each rnf port, indicates if the port is populated with a snoop responding device or not.

The input value is a string, for example 0xffff or ffff.

Type: string

Default value: "0"

acchannelen\_rni



Note

DEPRECATED: Will be removed after FM 11.18. Use rni\_sci\_enable instead.

For each rni port, indicates if the port is populated with a dvm responding device or not.

The input value is a string, for example `0xffff` or `ffff`.

Type: `string`

Default value: `"0"`

### **cache\_state\_modelled**

Model the cache state.

Type: `bool`

Default value: `false`

### **debug\_force\_snoop**

The CMN\_S3 interconnect will normally start with snooping disabled.

The parameter `rnf_sci_enable` and `rni_sci_enable` determines which connections are managed by the System Coherency Interface.

For connections that are managed by the System Coherency Interface, then they will look after themselves for entering and leaving the coherency domain and this parameter has no effect for those ports.

However, for connections that are *not* managed by the System Coherency Interface, this parameter enables the upstream system of a port to be snooped if the upstream is not in reset and if `acchannelen_rnf` allows it.



This parameter is for debug purpose only

Do not use this parameter instead of correctly configuring CMN.

---

Type: `bool`

Default value: `false`

### **dmc\_periphbase**

Value for DMC\_PERIPHBASE.

Type: `uint64_t`

Default value: `0xffffffffffffffff`

### **dsu\_periphbase**

Value for DSU\_PERIPHBASE.

Type: `uint64_t`

Default value: `0xffffffffffffffff`

### **enable\_a4s**

Enables A4S ports for GIC multi-chip routing.

Type: `bool`

Default value: `false`

### **enable\_logger**

Enable PVBUSLoggers for the downstream ports in the CMN model.

Type: `bool`

Default value: `false`

### **enable\_ras**

Enables RAS. There is an impact on performance when RAS is enabled.

Type: `bool`

Default value: `false`

### **enable\_rnsam\_to\_hnf\_wider\_hash**

Enable support of wider hash for the RNSAM to HNF communication.

If this variable enabled, then `bits[47:6]` from PA used in hashing function.

By default it is `[47:12]`.

Type: `bool`

Default value: `false`

### **force\_rnsam\_internal**

Force all RNSAMs to be internal independently of the mesh topology.

Type: `bool`

Default value: `true`

### **hnf\_mpam\_idr\_override**

Set to override `hnf_mpam_idr[31:24]` value. Bit[28] is Reserved and is ignored.

Type: `uint64_t`

Default value: `0`

**legacy\_tz\_en**

When set: Root registers accessible from Secure.

Realm Registers accessible from Non-Secure.

RCR are **RAZ/WI**.

Type: `bool`

Default value: `false`

**mesh\_config\_file**

Name of a file containing mesh placement of CMN\_S3 components.

Type: `string`

Default value: `""`

**mvp\_axu\_periphbase**

Value for MXP\_AXU\_PERIPHBASE.

Type: `uint64_t`

Default value: `0xffffffffffffffff`

**partner\_param0**

Partner Param.

Type: `uint64_t`

Default value: `0`

**periphbase**

Value for PERIPHBASE. Bits [27:0] are treated 0.

Type: `uint64_t`

Default value: `0x20000000`

**print\_cm\_n\_ccix\_config**

Print information about the CCIX configuration.

Type: `bool`

Default value: `false`

**print\_cm\_n\_config**

Print the mesh topology and children pointers acquired from the YML file.

Type: `bool`

Default value: `false`

### **register\_traces\_for\_ccg\_apb\_accesses**

---



Will be removed when enhancement SDDKW-74284 is done.

---

Intended for use with trace plugins.

#### **true**

registers traces to CCG register accesses through CCG APB interface.

#### **false**

registers traces to CMN register accesses through all other interfaces (eg RN nodes).

Type: `bool`

Default value: `false`

### **revision**

Component revision.

Currently supports r0, r1, r2.

Type: `string`

Default value: `"r0p0"`

### **rnf\_sci\_enable**

For each rnf port, indicates if the port is managed by System Coherency Interface for coherency domain entry/exit

#### **1**

Managed by SCI

#### **0**

Managed by Software

The input value is a string, for example `0xffff` or `ffff`.

Type: `string`

Default value: `"0x0"`

**rni\_sci\_enable**

For each rni port, indicates if the port is managed by System Coherency Interface for coherency domain entry/exit

**1**

Managed by SCI

**0**

Managed by Software

The input value is a string, for example `0xffff` or `ffff`.

Type: `string`

Default value: `"0x0"`

**show\_banner**

Show component banner:

**0**

supress entire banner

**1**

suppress config file

**2+**

show full banner.

Type: `uint64_t`

Default value: 2

**skip\_cmn\_config\_check**

Skip any topology configuration checks. The maximum number of devices per type not verified.

Type: `bool`

Default value: `false`

**use\_yaml\_periphbase**

Use yaml param `CFGM_PERIPHBASE_PARAM` to specify periphbase address.

If false, model parameter `periphbase` will be used.

Type: `bool`

Default value: `false`

**yaml\_has\_node\_addresses**

Does the top-level YML file describe node-addresses ?.

Type: `bool`

Default value: `false`

## 3.145 CMSDK\_Timer

Defined in `LISA/CMSDK_Timer.lisa`.

### About CMSDK\_Timer

ARM Timer Module.

### Iris and MTI instances for CMSDK\_Timer

This model has the following Iris instances:

Name	Instance type
<code>CMSDK_Timer</code>	<code>CMSDK_Timer</code>
<code>CMSDK_Timer.busslave</code>	<code>PVBusSlave</code>
<code>CMSDK_Timer.clk_div</code>	<code>ClockDivider</code>
<code>CMSDK_Timer.counter</code>	<code>CounterModule</code>
<code>CMSDK_Timer.counter.bussubordinate</code>	<code>PVBusSlave</code>

This model has the following MTI trace components:

Name	Component type
<code>CMSDK_Timer.busslave</code>	<code>PVBusSlave</code>
<code>CMSDK_Timer.clk_div</code>	<code>ClockDivider</code>
<code>CMSDK_Timer.counter.bussubordinate</code>	<code>PVBusSlave</code>

### Ports for CMSDK\_Timer

Port	Direction	Protocol	Description
<code>clock</code>	slave	<code>ClockSignal</code>	-
<code>irq_out</code>	master	<code>Signal</code>	-
<code>pvbus</code>	slave	<code>PVBus</code>	-

### Parameters for CMSDK\_Timer

#### `clk_div.div`

Clock Rate Divider.

Type: `uint64_t`

Default value: 1

**clk\_div.mul**  
Clock Rate Multiplier.

Type: uint64\_t  
Default value: 1

**diagnostics**  
Diagnostics.

Type: uint32\_t  
Default value: 0

### 3.146 Clock2SystemC

Defined in `examples/SystemCExport/Bridges/Clock2SystemC.lisa`.

**About Clock2SystemC**  
Clock to SystemC Converter.

**Iris and MTI instances for Clock2SystemC**  
This model has the following Iris instances:

Name	Instance type
Clock2SystemC	Clock2SystemC

#### Ports for Clock2SystemC

Port	Direction	Protocol	Description
clk_in	slave	ClockSignal	-
current_ticks_s	slave	AMBAPVValueState64	-
get_clock_s	slave	AMBAPVValueState64	-
rate_in_Hz_s	slave	AMBAPVValueState64	-
set_clock_m	master	AMBAPVValue64	-

**Parameters for Clock2SystemC**  
This component does not have any parameters.



### 3.147 ClockDivider

Defined in `LISA/ClockDivider.lisa`.

#### About ClockDivider

This component uses a configurable ratio to convert the `clockSignal` rate at its input to a new `clockSignal` rate at its output. Changes to the input rate or ratio take effect immediately and clocking components dependent on the output rate continue counting at the new rate.

This component does not normally incur a runtime performance cost. However, reprogramming the clock rate causes all related clocks and timers to be recalculated.



MasterClock is a 1 Hz clock. If the CPU clock frequency is not set to a realistic value, unpredictable behavior might occur, for example the simulation might freeze.

#### Iris and MTI instances for ClockDivider

This model has the following Iris instances:

Name	Instance type
ClockDivider	ClockDivider

This model has the following MTI trace components:

Name	Component type
ClockDivider	ClockDivider

#### Ports for ClockDivider

Port	Direction	Protocol	Description
clk_in	slave	ClockSignal	Input clock signal, coming from a MasterClock or another ClockDivider.
clk_out	master	ClockSignal	Clock signal generated by this ClockDivider.
rate	slave	ClockRateControl	Permits you to dynamically change the clock divider ratio.

#### Parameters for ClockDivider

##### div

Clock Rate Divider. This parameter is not exposed via Iris and can only be set in LISA.

Type: `uint64_t`

Default value: 1

##### mul

Clock Rate Multiplier. This parameter is not exposed via Iris and can only be set in LISA.

Type: uint64\_t

Default value: 1

## 3.148 ClockGate

Defined in LISA/ClockGate.lisa.

### About ClockGate

Clock gate for dis/enabling the clock.

### Iris and MTI instances for ClockGate

This model has the following Iris instances:

Name	Instance type
ClockGate	ClockGate
ClockGate.divider	ClockDivider

This model has the following MTI trace components:

Name	Component type
ClockGate.divider	ClockDivider

### Ports for ClockGate

Port	Direction	Protocol	Description
clk_enable	slave	Signal	-
clk_in	slave	ClockSignal	-
clk_out	master	ClockSignal	-
halt	master	Signal	-

### Parameters for ClockGate

#### diagnostics

Diagnostics.

Type: uint32\_t

Default value: 0

#### divider.div

Clock Rate Divider.

Type: uint64\_t

Default value: 1

**divider.mul**

Clock Rate Multiplier.

Type: uint64\_t

Default value: 1

3.149 ClockRateConversion

Defined in examples/SystemCEExport/Bridges/ClockRateConversion.lisa.

About ClockRateConversion

ClockRateControl to rate in Hz (Value\_64) Converter.

Iris and MTI instances for ClockRateConversion

This model has the following Iris instances:

Name	Instance type
ClockRateConversion	ClockRateConversion
ClockRateConversion.clk_divX (where X = 0-3)	ClockDivider

This model has the following MTI trace components:

Name	Component type
ClockRateConversion.clk_divX (where X = 0-3)	ClockDivider

Ports for ClockRateConversion

Port	Direction	Protocol	Description
clock	slave	ClockSignal	-
rate_ctrl	slave	ClockRateControl	-
rate_hz	master	Value_64	-

Parameters for ClockRateConversion

This component does not have any parameters.

## 3.150 ClockSelector

Defined in `LISA/ClockSelector.lisa`.

### About ClockSelector

ClockSignal Selector.

### Iris and MTI instances for ClockSelector

This model has the following Iris instances:

Name	Instance type
<code>ClockSelector</code>	<a href="#">ClockSelector</a>
<code>ClockSelector.clkdivX</code> (where $X = 0-10$ )	<a href="#">ClockDivider</a>
<code>ClockSelector.clkdivider</code>	<a href="#">ClockDivider</a>

This model has the following MTI trace components:

Name	Component type
<code>ClockSelector.clkdivX</code> (where $X = 0-10$ )	<a href="#">ClockDivider</a>
<code>ClockSelector.clkdivider</code>	<a href="#">ClockDivider</a>

### Ports for ClockSelector

Port	Direction	Protocol	Description
<code>clk_in</code>	slave	<a href="#">ClockSignal</a>	-
<code>clk_out</code>	master	<a href="#">ClockSignal</a>	-
<code>clk_sel_num</code>	slave	<a href="#">Value</a>	-
<code>clk_sel</code>	slave	<a href="#">Signal</a>	-

### Parameters for ClockSelector

#### **clkdiv0.div**

Clock Rate Divider.

Type: `uint64_t`

Default value: 1

#### **clkdiv0.mul**

Clock Rate Multiplier.

Type: `uint64_t`

Default value: 1

**clkdiv1.div**

Clock Rate Divider.

Type: uint64\_t

Default value: 1

**clkdiv1.mul**

Clock Rate Multiplier.

Type: uint64\_t

Default value: 1

**clkdiv10.div**

Clock Rate Divider.

Type: uint64\_t

Default value: 1

**clkdiv10.mul**

Clock Rate Multiplier.

Type: uint64\_t

Default value: 1

**clkdiv2.div**

Clock Rate Divider.

Type: uint64\_t

Default value: 1

**clkdiv2.mul**

Clock Rate Multiplier.

Type: uint64\_t

Default value: 1

**clkdiv3.div**

Clock Rate Divider.

Type: uint64\_t

Default value: 1

**clkdiv3.mul**

Clock Rate Multiplier.

Type: uint64\_t

Default value: 1

**clkdiv4.div**

Clock Rate Divider.

Type: uint64\_t

Default value: 1

**clkdiv4.mul**

Clock Rate Multiplier.

Type: uint64\_t

Default value: 1

**clkdiv5.div**

Clock Rate Divider.

Type: uint64\_t

Default value: 1

**clkdiv5.mul**

Clock Rate Multiplier.

Type: uint64\_t

Default value: 1

**clkdiv6.div**

Clock Rate Divider.

Type: uint64\_t

Default value: 1

**clkdiv6.mul**

Clock Rate Multiplier.

Type: uint64\_t

Default value: 1

**clkdiv7.div**

Clock Rate Divider.

Type: uint64\_t

Default value: 1

**clkdiv7.mul**

Clock Rate Multiplier.

Type: uint64\_t

Default value: 1

**clkdiv8.div**

Clock Rate Divider.

Type: uint64\_t

Default value: 1

**clkdiv8.mul**

Clock Rate Multiplier.

Type: uint64\_t

Default value: 1

**clkdiv9.div**

Clock Rate Divider.

Type: uint64\_t

Default value: 1

**clkdiv9.mul**

Clock Rate Multiplier.

Type: uint64\_t

Default value: 1

**clkdivider.div**

Clock Rate Divider.

Type: uint64\_t

Default value: 1

**clkdivider.mul**

Clock Rate Multiplier.

Type: uint64\_t

Default value: 1

**diagnostics**

Diagnostics.

Type: uint32\_t

Default value: 0

3.151 ClockSignal2SC\_ClockSignal

Defined in examples/SystemCExport/Bridges/ClockSignal2SC\_ClockSignal.lisa.

About ClockSignal2SC\_ClockSignal

ClockSignal to SystemC ClockSignal converter.

Iris and MTI instances for ClockSignal2SC\_ClockSignal

This model has the following Iris instances:

Name	Instance type
ClockSignal2SC_ClockSignal	ClockSignal2SC_ClockSignal

Ports for ClockSignal2SC\_ClockSignal

Port	Direction	Protocol	Description
clk_out	slave	ClockSignal	-
sc_clk_out	master	SC_ClockSignal	-

Parameters for ClockSignal2SC\_ClockSignal

This component does not have any parameters.

3.152 ClockTimer

Defined in LISA/ClockTimer.lisa.

About ClockTimer

When the countdown reaches zero, the timer can send a signal to another component. That component can return a value that causes the timer to start counting down again.



Setting up a timer is very efficient, and no host processing time is used while a counter is counting down: when a timer is started, the scheduler precomputes the finish time.

See `ClockSignalProtocol.lisa` and `CounterDivider.lisa` for more details of the scheduler system. See `TimerControlProtocol.lisa` and `TimerCallbackProtocol.lisa` for the methods used to control a timer and to handle the signal when the countdown is complete.

To use a `ClockTimer`:

1. Connect the `ClockTimer`'s `timer_callback` port to a slave port that implements the `signal()` behaviour.
2. Connect a clock signal to the `clk_in` port.
3. Use the `timer_control` port to start the timer counting down for a given number of ticks.

Ports for `ClockTimer`

Port	Direction	Protocol	Description
<code>clk_in</code>	slave	<a href="#">ClockSignal</a>	Determines the tick rate of the timer.
<code>timer_callback</code>	master	<a href="#">TimerCallback</a>	Port on which a signal is sent after the number of scheduled ticks has elapsed.
<code>timer_control</code>	slave	<a href="#">TimerControl</a>	Permits the timer to be set, canceled and queried.

Parameters for `ClockTimer`

This component does not have any parameters.

3.153 `ClockTimer64`

Defined in `LISA/ClockTimer64.lisa`.

About `ClockTimer64`

When the countdown reaches zero, the timer can send a signal to another component. That component can return a value that causes the timer to start counting down again.

Setting up a timer is very efficient, and no host processing time is used while a counter is counting down. When a timer is started, the scheduler precomputes the finish time.

This version of the timer provides 64-bit resolution.

See `ClockSignalProtocol64.lisa` and `CounterDivider.lisa` for more details of the scheduler system. See `TimerControlProtocol64.lisa` and `TimerCallbackProtocol.lisa` for the methods used to control a timer and to handle the signal when the countdown is complete.

To use a `ClockTimer64`:

1. Connect the `ClockTimer64`'s `timer_callback` port to a slave port that implements the `signal()` behaviour.
2. Connect a clock signal to the `clk_in` port.

- Use the `timer_control` port to start the timer counting down for a given number of ticks.

### Ports for ClockTimer64

Port	Direction	Protocol	Description
<code>clk_in</code>	slave	<a href="#">ClockSignal</a>	Determines the tick rate of the timer.
<code>timer_callback</code>	master	<a href="#">TimerCallback64</a>	Port on which a signal is sent after the number of scheduled ticks has elapsed.
<code>timer_control</code>	slave	<a href="#">TimerControl64</a>	Permits the timer to be set, canceled and queried.

### Parameters for ClockTimer64

This component does not have any parameters.

## 3.154 ClockTimerThread

Defined in `LISA/ClockTimerThread.lisa`.

### About ClockTimerThread

A `ClockTimerThread(64)` is a drop-in replacement for a `ClockTimer(64)` component. The main difference to the `clockTimer` component is that the `clockTimerThread` runs the `signal()` callback from a proper scheduler thread. This means that the `signal()` function may directly or indirectly invoke `wait()` functions to wait for time or events. This is not allowed for the `clockTimer` component which does not use a thread.

Components that issue bus transactions from within the timer `signal()` callback must use `ClockTimerThread(64)` rather than `clockTimer(64)`.

### Iris and MTI instances for ClockTimerThread

This model has the following Iris instances:

Name	Instance type
<code>ClockTimerThread</code>	<a href="#">ClockTimerThread</a>
<code>ClockTimerThread.timer</code>	<a href="#">ClockTimerThread64</a>
<code>ClockTimerThread.timer.thread</code>	<a href="#">SchedulerThread</a>
<code>ClockTimerThread.timer.thread_event</code>	<a href="#">SchedulerThreadEvent</a>

### Ports for ClockTimerThread

Port	Direction	Protocol	Description
<code>clk_in</code>	slave	<a href="#">ClockSignal</a>	Determines the tick rate of the timer.
<code>timer_callback</code>	master	<a href="#">TimerCallback</a>	Port on which a signal is sent after the number of scheduled ticks has elapsed.
<code>timer_control</code>	slave	<a href="#">TimerControl</a>	Permits the timer to be set, canceled and queried.

### Parameters for ClockTimerThread

This component does not have any parameters.

## 3.155 ClockTimerThread64

Defined in `LISA/ClockTimerThread64.lisa`.

### About ClockTimerThread64

A `ClockTimerThread(64)` is a drop-in replacement for a `ClockTimer(64)` component. The main difference to the `clockTimer` component is that the `clockTimerThread` runs the `signal()` callback from a proper scheduler thread. This means that the `signal()` function may directly or indirectly invoke `wait()` functions to wait for time or events. This is not allowed for the `clockTimer` component which does not use a thread.

Components that issue bus transactions from within the timer `signal()` callback must use `ClockTimerThread(64)` rather than `clockTimer(64)`.

### Iris and MTI instances for ClockTimerThread64

This model has the following Iris instances:

Name	Instance type
<code>ClockTimerThread64</code>	<a href="#">ClockTimerThread64</a>
<code>ClockTimerThread64.thread</code>	<a href="#">SchedulerThread</a>
<code>ClockTimerThread64.thread_event</code>	<a href="#">SchedulerThreadEvent</a>

### Ports for ClockTimerThread64

Port	Direction	Protocol	Description
<code>clk_in</code>	slave	<a href="#">ClockSignal</a>	-
<code>timer_callback</code>	master	<a href="#">TimerCallback64</a>	-
<code>timer_control</code>	slave	<a href="#">TimerControl64</a>	-

### Parameters for ClockTimerThread64

This component does not have any parameters.

## 3.156 Clock\_Multiplexer

Defined in `LISA/Multiplexer.lisa`.

### About Clock\_Multiplexer

Clock Multiplexer.

### Iris and MTI instances for Clock\_Multiplexer

This model has the following Iris instances:

Name	Instance type
Clock_Multiplexer	Clock_Multiplexer

### Ports for Clock\_Multiplexer

Port	Direction	Protocol	Description
cur_clkssel	master	Value	-
input	slave	ClockSignal	-
output	master	ClockSignal	-
selector	slave	Value	-
stopclk_nominal	slave	Value	-

### Parameters for Clock\_Multiplexer

#### diagnostics

Diagnostics.

Type: uint8\_t

Default value: 2

## 3.157 ClusterClockControl

Defined in LISA/ClusterClockControl.lisa.

### About ClusterClockControl

Cluster clock control allows input selection, rate control and gating.

### Iris and MTI instances for ClusterClockControl

This model has the following Iris instances:

Name	Instance type
ClusterClockControl	ClusterClockControl
ClusterClockControl.clkGate	ClockGate
ClusterClockControl.clkGate.divider	ClockDivider
ClusterClockControl.clkSelector	ClockSelector
ClusterClockControl.clkSelector.clkdivX (where X = 0-10)	ClockDivider
ClusterClockControl.clkSelector.clkdivider	ClockDivider
ClusterClockControl.refClkDiv	ClockDivider
ClusterClockControl.sysClkDiv	ClockDivider
ClusterClockControl.xClkDiv	ClockDivider

This model has the following MTI trace components:

Name	Component type
ClusterClockControl.clkGate.divider	<a href="#">ClockDivider</a>
ClusterClockControl.clkSelector.clkdivX (where X = 0-10)	<a href="#">ClockDivider</a>
ClusterClockControl.clkSelector.clkdivider	<a href="#">ClockDivider</a>
ClusterClockControl.refClkDiv	<a href="#">ClockDivider</a>
ClusterClockControl.sysClkDiv	<a href="#">ClockDivider</a>
ClusterClockControl.xClkDiv	<a href="#">ClockDivider</a>

### Ports for ClusterClockControl

Port	Direction	Protocol	Description
clk_out	master	<a href="#">ClockSignal</a>	-
clkDivExt	slave	<a href="#">ClockRateControl</a>	-
clkDivSys	slave	<a href="#">ClockRateControl</a>	-
clkEnable	slave	<a href="#">Signal</a>	-
clkSel	slave	<a href="#">Value</a>	-
halt	master	<a href="#">Signal</a>	-
refClk_in	slave	<a href="#">ClockSignal</a>	-
sysClk_in	slave	<a href="#">ClockSignal</a>	-
xClk_in	slave	<a href="#">ClockSignal</a>	-

### Parameters for ClusterClockControl

#### **clkGate.diagnostics**

Diagnostics.

Type: `uint32_t`

Default value: 0

#### **clkGate.divider.div**

Clock Rate Divider.

Type: `uint64_t`

Default value: 1

#### **clkGate.divider.mul**

Clock Rate Multiplier.

Type: `uint64_t`

Default value: 1

**clkSelector.clkdiv0.div**

Clock Rate Divider.

Type: uint64\_t

Default value: 1

**clkSelector.clkdiv0.mul**

Clock Rate Multiplier.

Type: uint64\_t

Default value: 1

**clkSelector.clkdiv1.div**

Clock Rate Divider.

Type: uint64\_t

Default value: 1

**clkSelector.clkdiv1.mul**

Clock Rate Multiplier.

Type: uint64\_t

Default value: 1

**clkSelector.clkdiv10.div**

Clock Rate Divider.

Type: uint64\_t

Default value: 1

**clkSelector.clkdiv10.mul**

Clock Rate Multiplier.

Type: uint64\_t

Default value: 1

**clkSelector.clkdiv2.div**

Clock Rate Divider.

Type: uint64\_t

Default value: 1

**clkSelector.clkdiv2.mul**

Clock Rate Multiplier.

Type: uint64\_t

Default value: 1

**clkSelector.clkdiv3.div**

Clock Rate Divider.

Type: uint64\_t

Default value: 1

**clkSelector.clkdiv3.mul**

Clock Rate Multiplier.

Type: uint64\_t

Default value: 1

**clkSelector.clkdiv4.div**

Clock Rate Divider.

Type: uint64\_t

Default value: 1

**clkSelector.clkdiv4.mul**

Clock Rate Multiplier.

Type: uint64\_t

Default value: 1

**clkSelector.clkdiv5.div**

Clock Rate Divider.

Type: uint64\_t

Default value: 1

**clkSelector.clkdiv5.mul**

Clock Rate Multiplier.

Type: uint64\_t

Default value: 1

**clkSelector.clkdiv6.div**

Clock Rate Divider.

Type: uint64\_t

Default value: 1

**clkSelector.clkdiv6.mul**

Clock Rate Multiplier.

Type: uint64\_t

Default value: 1

**clkSelector.clkdiv7.div**

Clock Rate Divider.

Type: uint64\_t

Default value: 1

**clkSelector.clkdiv7.mul**

Clock Rate Multiplier.

Type: uint64\_t

Default value: 1

**clkSelector.clkdiv8.div**

Clock Rate Divider.

Type: uint64\_t

Default value: 1

**clkSelector.clkdiv8.mul**

Clock Rate Multiplier.

Type: uint64\_t

Default value: 1

**clkSelector.clkdiv9.div**

Clock Rate Divider.

Type: uint64\_t

Default value: 1



**clkSelector.clkdiv9.mul**

Clock Rate Multiplier.

Type: uint64\_t

Default value: 1

**clkSelector.clkdivider.div**

Clock Rate Divider.

Type: uint64\_t

Default value: 1

**clkSelector.clkdivider.mul**

Clock Rate Multiplier.

Type: uint64\_t

Default value: 1

**clkSelector.diagnostics**

Diagnostics.

Type: uint32\_t

Default value: 0

**refClkDiv.div**

Clock Rate Divider.

Type: uint64\_t

Default value: 1

**refClkDiv.mul**

Clock Rate Multiplier.

Type: uint64\_t

Default value: 1

**sysClkDiv.div**

Clock Rate Divider.

Type: uint64\_t

Default value: 1

**sysClkDiv.mul**

Clock Rate Multiplier.

Type: uint64\_t

Default value: 1

**xClkDiv.div**

Clock Rate Divider.

Type: uint64\_t

Default value: 1

**xClkDiv.mul**

Clock Rate Multiplier.

Type: uint64\_t

Default value: 1

## 3.158 Cluster\_Temperature\_Sensor

Defined in LISA/Cluster\_Temperature\_Sensor.lisa.

### About Cluster\_Temperature\_Sensor

Component to calculate the temperature value of all cores in a cluster.

### Iris and MTI instances for Cluster\_Temperature\_Sensor

This model has the following Iris instances:

Name	Instance type
Cluster_Temperature_Sensor	Cluster_Temperature_Sensor

### Ports for Cluster\_Temperature\_Sensor

Port	Direction	Protocol	Description
cluster_powerdown_in	slave	Signal	-
core_powerdown_in	slave	Signal	-
core_state_in	slave	ValueState	-
core_ticks_in	slave	InstructionCount	-
freq_in	slave	ValueState	-
temperature_out	master	ValueState	-

## Parameters for Cluster\_Temperature\_Sensor

### **MAX\_FREQ**

Maximum frequency at which each core can run.

Type: `uint64_t`

Default value: 2000000000

### **NUM\_CORES**

Number of cores per cluster.

Type: `uint32_t`

Default value: 4

### **diagnostics**

Diagnostics.

Type: `uint32_t`

Default value: 0

### **tempCoeff\_A**

Temperature Coefficient.

Type: `string`

Default value: "0.5"

### **tempCoeff\_B**

Temperature Coefficient.

Type: `string`

Default value: "0.5"

### **tempCoeff\_K**

Temperature Coefficient.

Type: `uint32_t`

Default value: 50

### **tempCoeff\_TAMB**

Temperature Coefficient.

Type: uint32\_t

Default value: 20

## 3.159 CombinedMessagingUnit

Defined in LISA/CombinedMessagingUnit.lisa.

### Changes in 11.31.15

The following parameters were added:

- host\_to\_local.a\_to\_b\_v3.receiver\_legacy\_tz\_en
- host\_to\_local.a\_to\_b\_v3.sender\_legacy\_tz\_en
- host\_to\_local.a\_to\_b\_v3.support\_receiver\_rme
- host\_to\_local.a\_to\_b\_v3.support\_receiver\_tze
- host\_to\_local.a\_to\_b\_v3.support\_sender\_rme
- host\_to\_local.a\_to\_b\_v3.support\_sender\_tze
- local\_to\_host.a\_to\_b\_v3.receiver\_legacy\_tz\_en
- local\_to\_host.a\_to\_b\_v3.sender\_legacy\_tz\_en
- local\_to\_host.a\_to\_b\_v3.support\_receiver\_rme
- local\_to\_host.a\_to\_b\_v3.support\_receiver\_tze
- local\_to\_host.a\_to\_b\_v3.support\_sender\_rme
- local\_to\_host.a\_to\_b\_v3.support\_sender\_tze

### About CombinedMessagingUnit

CMU - Combined MHU monolithic block.

### Iris and MTI instances for CombinedMessagingUnit

This model has the following Iris instances:

Name	Instance type
CombinedMessagingUnit	CombinedMessagingUnit
CombinedMessagingUnit.host_to_local	MessageHandlingUnit
CombinedMessagingUnit.host_to_local.a_to_b_v2	MessageHandlingUnitV2
CombinedMessagingUnit.host_to_local.a_to_b_v3	MessageHandlingUnitV3
CombinedMessagingUnit.host_to_local.version_mapper_a_to_b_rec	PVBusMapper
CombinedMessagingUnit.host_to_local.version_mapper_a_to_b_snd	PVBusMapper
CombinedMessagingUnit.host_to_local_rcv_log	PVBusLogger
CombinedMessagingUnit.host_to_local_rcv_log.mapper	PVBusMapper
CombinedMessagingUnit.host_to_local_snd_log	PVBusLogger

Name	Instance type
CombinedMessagingUnit.host_to_local_snd_log.mapper	PVBusMapper
CombinedMessagingUnit.irq_rcv_combined_host_log	SignalLogger
CombinedMessagingUnit.irq_rcv_combined_local_log	SignalLogger
CombinedMessagingUnit.irq_snd_combined_host_log	SignalLogger
CombinedMessagingUnit.irq_snd_combined_local_log	SignalLogger
CombinedMessagingUnit.local_to_host	MessageHandlingUnit
CombinedMessagingUnit.local_to_host.a_to_b_v2	MessageHandlingUnitV2
CombinedMessagingUnit.local_to_host.a_to_b_v3	MessageHandlingUnitV3
CombinedMessagingUnit.local_to_host.version_mapper_a_to_b_rec	PVBusMapper
CombinedMessagingUnit.local_to_host.version_mapper_a_to_b_snd	PVBusMapper
CombinedMessagingUnit.local_to_host_rcv_log	PVBusLogger
CombinedMessagingUnit.local_to_host_rcv_log.mapper	PVBusMapper
CombinedMessagingUnit.local_to_host_snd_log	PVBusLogger
CombinedMessagingUnit.local_to_host_snd_log.mapper	PVBusMapper

This model has the following MTI trace components:

Name	Component type
CombinedMessagingUnit.host_to_local.a_to_b_v2	MessageHandlingUnitV2
CombinedMessagingUnit.host_to_local.a_to_b_v3	MessageHandlingUnitV3
CombinedMessagingUnit.host_to_local.version_mapper_a_to_b_rec	PVBusMapper
CombinedMessagingUnit.host_to_local.version_mapper_a_to_b_snd	PVBusMapper
CombinedMessagingUnit.host_to_local_rcv_log	PVBusLogger
CombinedMessagingUnit.host_to_local_rcv_log.mapper	PVBusMapper
CombinedMessagingUnit.host_to_local_snd_log	PVBusLogger
CombinedMessagingUnit.host_to_local_snd_log.mapper	PVBusMapper
CombinedMessagingUnit.irq_rcv_combined_host_log	SignalLogger
CombinedMessagingUnit.irq_rcv_combined_local_log	SignalLogger
CombinedMessagingUnit.irq_snd_combined_host_log	SignalLogger
CombinedMessagingUnit.irq_snd_combined_local_log	SignalLogger
CombinedMessagingUnit.local_to_host.a_to_b_v2	MessageHandlingUnitV2
CombinedMessagingUnit.local_to_host.a_to_b_v3	MessageHandlingUnitV3
CombinedMessagingUnit.local_to_host.version_mapper_a_to_b_rec	PVBusMapper
CombinedMessagingUnit.local_to_host.version_mapper_a_to_b_snd	PVBusMapper
CombinedMessagingUnit.local_to_host_rcv_log	PVBusLogger
CombinedMessagingUnit.local_to_host_rcv_log.mapper	PVBusMapper
CombinedMessagingUnit.local_to_host_snd_log	PVBusLogger
CombinedMessagingUnit.local_to_host_snd_log.mapper	PVBusMapper

## Ports for CombinedMessagingUnit

Port	Direction	Protocol	Description
irq_rcv_combined_host	master	Signal	-
irq_rcv_combined_local	master	Signal	-
irq_snd_combined_host	master	Signal	-
irq_snd_combined_local	master	Signal	-
pvbus_s_rcv_host	slave	PVBus	-
pvbus_s_rcv_local	slave	PVBus	-
pvbus_s_snd_host	slave	PVBus	-
pvbus_s_snd_local	slave	PVBus	-
reset_in	slave	Signal	-

## Parameters for CombinedMessagingUnit

### **NUM\_DB\_CH**

Number of doorbell channels.

Type: `uint32_t`

Default value: 1

### **NUM\_FAST\_CH**

Number of fast channels.

Type: `uint32_t`

Default value: 1

### **diagnostics**

Diagnostics 0==FATAL\_ERROR -> 4==DEBUG.

Type: `uint8_t`

Default value: 2

### **fast\_ch\_group\_int\_enable**

Fast Channel group interrupts enable, default=false.

Type: `bool`

Default value: `false`

### **fast\_ch\_n\_per\_group**

Fast Channel num channels per group, default=1.

Type: `uint32_t`

Default value: 1

**fast\_ch\_num\_groups**

Fast Channel num of groups, default=1.

Type: uint32\_t

Default value: 1

**fast\_ch\_word\_size**

Fast Channel word size 32bit or 64bit, default=32.

Type: uint32\_t

Default value: 32

**host\_to\_local.NUM\_DB\_CH**

Number of doorbell channels.

Type: uint32\_t

Default value: 1

**host\_to\_local.NUM\_FAST\_CH**

Number of fast channels.

Type: uint32\_t

Default value: 1

**host\_to\_local.a\_to\_b\_v2.NUM\_CH**

Number of device channels.

Type: uint32\_t

Default value: 1

**host\_to\_local.a\_to\_b\_v2.minor\_revision**

MHUv2 minor revision.

Type: uint32\_t

Default value: 0

**host\_to\_local.a\_to\_b\_v2.product\_id**

MHU part number.

Type: uint32\_t

Default value: 0

**host\_to\_local.a\_to\_b\_v3.NUM\_DB\_CH**

Number of doorbell channels.

Type: uint32\_t

Default value: 1

**host\_to\_local.a\_to\_b\_v3.NUM\_FAST\_CH**

Number of Fast Channels.

Type: uint32\_t

Default value: 1

**host\_to\_local.a\_to\_b\_v3.NUM\_FIFO\_CH**

Number of FIFO Channels.

Type: uint32\_t

Default value: 1

**host\_to\_local.a\_to\_b\_v3.auto\_op\_full**

AutoOp mode - AutoOp(min) == false, AutoOp(full) == true - default: AutoOp(min).

Type: bool

Default value: false

**host\_to\_local.a\_to\_b\_v3.diagnostics**

Diagnostics 0:FATAL 1:ERROR 2:WARNING 3:INFO 4:DEBUG, Default:WARNING(2).

Type: uint8\_t

Default value: 2

**host\_to\_local.a\_to\_b\_v3.fast\_ch\_group\_int\_enable**

Fast Channel group interrupts enable, default=false.

Type: bool

Default value: false

**host\_to\_local.a\_to\_b\_v3.fast\_ch\_n\_per\_group**

Fast Channel num channels per group, default=1.

Type: uint32\_t



Default value: 1

**host\_to\_local.a\_to\_b\_v3.fast\_ch\_num\_groups**

Fast Channel num of groups, default=1.

Type: uint32\_t

Default value: 1

**host\_to\_local.a\_to\_b\_v3.fast\_ch\_word\_size**

Fast Channel word size 32bit or 64bit, default=32.

Type: uint32\_t

Default value: 32

**host\_to\_local.a\_to\_b\_v3.fifo\_depth**

Depth of the FIFO = fifo\_depth + 1.

Type: uint16\_t

Default value: 4

**host\_to\_local.a\_to\_b\_v3.m16ba\_spt**

Mailbox 16 bit access support to FIFO registers.

Type: "bool"

Default value: 0

**host\_to\_local.a\_to\_b\_v3.m32ba\_spt**

Mailbox 32 bit access support to FIFO registers.

Type: "bool"

Default value: 1

**host\_to\_local.a\_to\_b\_v3.m64ba\_spt**

Mailbox 64 bit access support to FIFO registers.

Type: "bool"

Default value: 0

**host\_to\_local.a\_to\_b\_v3.m8ba\_spt**

Mailbox 8 bit access support to FIFO registers.

Type: "bool"

Default value: 0

**host\_to\_local.a\_to\_b\_v3.mhu\_arch\_beta01**

true = Aligns to MHUv3.beta01; false = Aligns to MHUv3.2.

Type: bool

Default value: false

**host\_to\_local.a\_to\_b\_v3.monolithic**

Monolithic or Distributed MHU - default: monolithic(true).

Type: bool

Default value: true

**host\_to\_local.a\_to\_b\_v3.p16ba\_spt**

Postbox 16 bit access support to FIFO registers.

Type: "bool"

Default value: 0

**host\_to\_local.a\_to\_b\_v3.p32ba\_spt**

Postbox 32 bit access support to FIFO registers.

Type: "bool"

Default value: 1

**host\_to\_local.a\_to\_b\_v3.p64ba\_spt**

Postbox 64 bit access support to FIFO registers.

Type: "bool"

Default value: 0

**host\_to\_local.a\_to\_b\_v3.p8ba\_spt**

Postbox 8 bit access support to FIFO registers.

Type: "bool"

Default value: 0

**host\_to\_local.a\_to\_b\_v3.receiver\_legacy\_tz\_en**

Receiver Legacy TrustZone Enable - default: false.

Type: bool

Default value: `false`

### **host\_to\_local.a\_to\_b\_v3.sender\_legacy\_tz\_en**

Sender Legacy TrustZone Enable - default: `false`.

Type: `bool`

Default value: `false`

### **host\_to\_local.a\_to\_b\_v3.support\_receiver\_rme**

Support RME functionality in receiver, set to true if the underlying platform supports RME.

Type: `bool`

Default value: `false`

### **host\_to\_local.a\_to\_b\_v3.support\_receiver\_tze**

Support TrustZone functionality in receiver, set to true if the underlying platform supports TrustZone.

Type: `bool`

Default value: `false`

### **host\_to\_local.a\_to\_b\_v3.support\_sender\_rme**

Support RME functionality in sender, set to true if the underlying platform supports RME.

Type: `bool`

Default value: `false`

### **host\_to\_local.a\_to\_b\_v3.support\_sender\_tze**

Support TrustZone functionality in sender, set to true if the underlying platform supports TrustZone.

Type: `bool`

Default value: `false`

### **host\_to\_local.diagnostics**

Diagnostics 0==FATAL\_ERROR -> 4==DEBUG.

Type: `uint8_t`

Default value: 2

**host\_to\_local.fast\_ch\_group\_int\_enable**

Fast Channel group interrupts enable, default=false.

Type: `bool`

Default value: `false`

**host\_to\_local.fast\_ch\_n\_per\_group**

Fast Channel num channels per group, default=1.

Type: `uint32_t`

Default value: 1

**host\_to\_local.fast\_ch\_num\_groups**

Fast Channel num of groups, default=1.

Type: `uint32_t`

Default value: 1

**host\_to\_local.fast\_ch\_word\_size**

Fast Channel word size 32bit or 64bit, default=32.

Type: `uint32_t`

Default value: 32

**host\_to\_local.major\_version**

MHU major version (default=2).

Type: `uint32_t`

Default value: 2

**host\_to\_local.mhu\_arch\_beta01**

true = Aligns to MHUv3.beta01; false = Aligns to MHUv3.2.

Type: `bool`

Default value: `false`

**host\_to\_local.minor\_version**

MHU minor version (default=1).

Type: `uint32_t`

Default value: 1

**host\_to\_local.product\_id**

MHU part number.

Type: `uint32_t`

Default value: `0`

**host\_to\_local\_rcv\_log.trace\_debug**

Enable tracing of debug transactions.

Type: `bool`

Default value: `false`

**host\_to\_local\_rcv\_log.trace\_snoops**

Enable tracing of ACE snoop requests.

Type: `bool`

Default value: `false`

**host\_to\_local\_snd\_log.trace\_debug**

Enable tracing of debug transactions.

Type: `bool`

Default value: `false`

**host\_to\_local\_snd\_log.trace\_snoops**

Enable tracing of ACE snoop requests.

Type: `bool`

Default value: `false`

**irq\_rcv\_combined\_host\_log.forward\_signal**

If true, trace signal and forward signal from 'in' to 'out'. If false, trace signal only without driving 'out' port.

Type: `bool`

Default value: `true`

**irq\_rcv\_combined\_local\_log.forward\_signal**

If true, trace signal and forward signal from 'in' to 'out'. If false, trace signal only without driving 'out' port.

Type: `bool`

Default value: `true`

#### **`irq_snd_combined_host_log.forward_signal`**

If true, trace signal and forward signal from 'in' to 'out'. If false, trace signal only without driving 'out' port.

Type: `bool`

Default value: `true`

#### **`irq_snd_combined_local_log.forward_signal`**

If true, trace signal and forward signal from 'in' to 'out'. If false, trace signal only without driving 'out' port.

Type: `bool`

Default value: `true`

#### **`local_to_host.NUM_DB_CH`**

Number of doorbell channels.

Type: `uint32_t`

Default value: 1

#### **`local_to_host.NUM_FAST_CH`**

Number of fast channels.

Type: `uint32_t`

Default value: 1

#### **`local_to_host.a_to_b_v2.NUM_CH`**

Number of device channels.

Type: `uint32_t`

Default value: 1

#### **`local_to_host.a_to_b_v2.minor_revision`**

MHUV2 minor revision.

Type: `uint32_t`

Default value: 0

**local\_to\_host.a\_to\_b\_v2.product\_id**

MHU part number.

Type: uint32\_t

Default value: 0

**local\_to\_host.a\_to\_b\_v3.NUM\_DB\_CH**

Number of doorbell channels.

Type: uint32\_t

Default value: 1

**local\_to\_host.a\_to\_b\_v3.NUM\_FAST\_CH**

Number of Fast Channels.

Type: uint32\_t

Default value: 1

**local\_to\_host.a\_to\_b\_v3.NUM\_FIFO\_CH**

Number of FIFO Channels.

Type: uint32\_t

Default value: 1

**local\_to\_host.a\_to\_b\_v3.auto\_op\_full**

AutoOp mode - AutoOp(min) == false, AutoOp(full) == true - default: AutoOp(min).

Type: bool

Default value: false

**local\_to\_host.a\_to\_b\_v3.diagnostics**

Diagnostics 0:FATAL 1:ERROR 2:WARNING 3:INFO 4:DEBUG, Default:WARNING(2).

Type: uint8\_t

Default value: 2

**local\_to\_host.a\_to\_b\_v3.fast\_ch\_group\_int\_enable**

Fast Channel group interrupts enable, default=false.

Type: bool

Default value: false

**local\_to\_host.a\_to\_b\_v3.fast\_ch\_n\_per\_group**

Fast Channel num channels per group, default=1.

Type: uint32\_t

Default value: 1

**local\_to\_host.a\_to\_b\_v3.fast\_ch\_num\_groups**

Fast Channel num of groups, default=1.

Type: uint32\_t

Default value: 1

**local\_to\_host.a\_to\_b\_v3.fast\_ch\_word\_size**

Fast Channel word size 32bit or 64bit, default=32.

Type: uint32\_t

Default value: 32

**local\_to\_host.a\_to\_b\_v3.fifo\_depth**

Depth of the FIFO = fifo\_depth + 1.

Type: uint16\_t

Default value: 4

**local\_to\_host.a\_to\_b\_v3.m16ba\_spt**

Mailbox 16 bit access support to FIFO registers.

Type: "bool"

Default value: 0

**local\_to\_host.a\_to\_b\_v3.m32ba\_spt**

Mailbox 32 bit access support to FIFO registers.

Type: "bool"

Default value: 1

**local\_to\_host.a\_to\_b\_v3.m64ba\_spt**

Mailbox 64 bit access support to FIFO registers.

Type: "bool"

Default value: 0



**local\_to\_host.a\_to\_b\_v3.m8ba\_spt**

Mailbox 8 bit access support to FIFO registers.

Type: "bool"

Default value: 0

**local\_to\_host.a\_to\_b\_v3.mhu\_arch\_beta01**

true = Aligns to MHUv3.beta01; false = Aligns to MHUv3.2.

Type: bool

Default value: false

**local\_to\_host.a\_to\_b\_v3.monolithic**

Monolithic or Distributed MHU - default: monolithic(true).

Type: bool

Default value: true

**local\_to\_host.a\_to\_b\_v3.p16ba\_spt**

Postbox 16 bit access support to FIFO registers.

Type: "bool"

Default value: 0

**local\_to\_host.a\_to\_b\_v3.p32ba\_spt**

Postbox 32 bit access support to FIFO registers.

Type: "bool"

Default value: 1

**local\_to\_host.a\_to\_b\_v3.p64ba\_spt**

Postbox 64 bit access support to FIFO registers.

Type: "bool"

Default value: 0

**local\_to\_host.a\_to\_b\_v3.p8ba\_spt**

Postbox 8 bit access support to FIFO registers.

Type: "bool"

Default value: 0

**local\_to\_host.a\_to\_b\_v3.receiver\_legacy\_tz\_en**

Receiver Legacy TrustZone Enable - default: false.

Type: bool

Default value: false

**local\_to\_host.a\_to\_b\_v3.sender\_legacy\_tz\_en**

Sender Legacy TrustZone Enable - default: false.

Type: bool

Default value: false

**local\_to\_host.a\_to\_b\_v3.support\_receiver\_rme**

Support RME functionality in receiver, set to true if the underlying platform supports RME.

Type: bool

Default value: false

**local\_to\_host.a\_to\_b\_v3.support\_receiver\_tze**

Support TrustZone functionality in receiver, set to true if the underlying platform supports TrustZone.

Type: bool

Default value: false

**local\_to\_host.a\_to\_b\_v3.support\_sender\_rme**

Support RME functionality in sender, set to true if the underlying platform supports RME.

Type: bool

Default value: false

**local\_to\_host.a\_to\_b\_v3.support\_sender\_tze**

Support TrustZone functionality in sender, set to true if the underlying platform supports TrustZone.

Type: bool

Default value: false

**local\_to\_host.diagnostics**

Diagnostics 0==FATAL\_ERROR -> 4==DEBUG.

Type: `uint8_t`

Default value: 2

### **`local_to_host.fast_ch_group_int_enable`**

Fast Channel group interrupts enable, default=false.

Type: `bool`

Default value: `false`

### **`local_to_host.fast_ch_n_per_group`**

Fast Channel num channels per group, default=1.

Type: `uint32_t`

Default value: 1

### **`local_to_host.fast_ch_num_groups`**

Fast Channel num of groups, default=1.

Type: `uint32_t`

Default value: 1

### **`local_to_host.fast_ch_word_size`**

Fast Channel word size 32bit or 64bit, default=32.

Type: `uint32_t`

Default value: 32

### **`local_to_host.major_version`**

MHU major version (default=2).

Type: `uint32_t`

Default value: 2

### **`local_to_host.mhu_arch_beta01`**

true = Aligns to MHUv3.beta01; false = Aligns to MHUv3.2.

Type: `bool`

Default value: `false`

### **`local_to_host.minor_version`**

MHU minor version (default=1).

Type: `uint32_t`

Default value: 1

### **`local_to_host.product_id`**

MHU part number.

Type: `uint32_t`

Default value: 0

### **`local_to_host_rcv_log.trace_debug`**

Enable tracing of debug transactions.

Type: `bool`

Default value: `false`

### **`local_to_host_rcv_log.trace_snoops`**

Enable tracing of ACE snoop requests.

Type: `bool`

Default value: `false`

### **`local_to_host_snd_log.trace_debug`**

Enable tracing of debug transactions.

Type: `bool`

Default value: `false`

### **`local_to_host_snd_log.trace_snoops`**

Enable tracing of ACE snoop requests.

Type: `bool`

Default value: `false`

### **`major_version`**

MHU major version (default=2).

Type: `uint32_t`

Default value: 2

### **`mhu_arch_beta01`**

true = Aligns to MHUv3.beta01; false = Aligns to MHUv3.2.

Type: `bool`

Default value: `false`

### **minor\_version**

MHU minor version (default=1).

Type: `uint32_t`

Default value: 1

### **product\_id**

MHU part number.

Type: `uint32_t`

Default value: 0

## 3.160 CombinedMessagingUnitAE

Defined in `LISA/CombinedMessagingUnitAE.lisa`.

### About CombinedMessagingUnitAE

CMU AE - Combined MHU320AE monolithic block.

### Iris and MTI instances for CombinedMessagingUnitAE

This model has the following Iris instances:

Name	Instance type
CombinedMessagingUnitAE	CombinedMessagingUnitAE
CombinedMessagingUnitAE.host_to_local	MHU320AE
CombinedMessagingUnitAE.host_to_local.MHU320AE FMU	mhu320ae_fmu
CombinedMessagingUnitAE.host_to_local_rcv_log	PVBusLogger
CombinedMessagingUnitAE.host_to_local_rcv_log.mapper	PVBusMapper
CombinedMessagingUnitAE.host_to_local_snd_log	PVBusLogger
CombinedMessagingUnitAE.host_to_local_snd_log.mapper	PVBusMapper
CombinedMessagingUnitAE.irq_rcv_combined_host_log	SignalLogger
CombinedMessagingUnitAE.irq_rcv_combined_local_log	SignalLogger
CombinedMessagingUnitAE.irq_snd_combined_host_log	SignalLogger
CombinedMessagingUnitAE.irq_snd_combined_local_log	SignalLogger
CombinedMessagingUnitAE.local_to_host	MHU320AE
CombinedMessagingUnitAE.local_to_host.MHU320AE FMU	mhu320ae_fmu
CombinedMessagingUnitAE.local_to_host_rcv_log	PVBusLogger

Name	Instance type
CombinedMessagingUnitAE.local_to_host_rcv_log.mapper	PVBusMapper
CombinedMessagingUnitAE.local_to_host_snd_log	PVBusLogger
CombinedMessagingUnitAE.local_to_host_snd_log.mapper	PVBusMapper

This model has the following MTI trace components:

Name	Component type
CombinedMessagingUnitAE.host_to_local	MessageHandlingUnitV3
CombinedMessagingUnitAE.host_to_local_rcv_log	PVBusLogger
CombinedMessagingUnitAE.host_to_local_rcv_log.mapper	PVBusMapper
CombinedMessagingUnitAE.host_to_local_snd_log	PVBusLogger
CombinedMessagingUnitAE.host_to_local_snd_log.mapper	PVBusMapper
CombinedMessagingUnitAE.irq_rcv_combined_host_log	SignalLogger
CombinedMessagingUnitAE.irq_rcv_combined_local_log	SignalLogger
CombinedMessagingUnitAE.irq_snd_combined_host_log	SignalLogger
CombinedMessagingUnitAE.irq_snd_combined_local_log	SignalLogger
CombinedMessagingUnitAE.local_to_host	MessageHandlingUnitV3
CombinedMessagingUnitAE.local_to_host_rcv_log	PVBusLogger
CombinedMessagingUnitAE.local_to_host_rcv_log.mapper	PVBusMapper
CombinedMessagingUnitAE.local_to_host_snd_log	PVBusLogger
CombinedMessagingUnitAE.local_to_host_snd_log.mapper	PVBusMapper

### Ports for CombinedMessagingUnitAE

Port	Direction	Protocol	Description
fmu_cri_out_host	master	Signal	-
fmu_cri_out_local	master	Signal	-
fmu_eri_out_host	master	Signal	-
fmu_eri_out_local	master	Signal	-
irq_rcv_combined_host	master	Signal	-
irq_rcv_combined_local	master	Signal	-
irq_snd_combined_host	master	Signal	-
irq_snd_combined_local	master	Signal	-
pvbus_s_rcv_host	slave	PVBus	-
pvbus_s_rcv_local	slave	PVBus	-
pvbus_s_snd_fmu_host	slave	PVBus	-
pvbus_s_snd_fmu_local	slave	PVBus	-
pvbus_s_snd_host	slave	PVBus	-
pvbus_s_snd_local	slave	PVBus	-
reset_in	slave	Signal	-

## Parameters for CombinedMessagingUnitAE

### **NUM\_DB\_CH\_H2L**

Number of doorbell channels Host to Local.

Type: `uint32_t`

Default value: 1

### **NUM\_DB\_CH\_L2H**

Number of doorbell channels Local to Host.

Type: `uint32_t`

Default value: 1

### **NUM\_FAST\_CH**

Number of fast channels.

Type: `uint32_t`

Default value: 1

### **diagnostics**

Diagnostics 0==FATAL\_ERROR -> 4==DEBUG.

Type: `uint8_t`

Default value: 2

### **fast\_ch\_group\_int\_enable**

Fast Channel group interrupts enable, default=false.

Type: `bool`

Default value: `false`

### **fast\_ch\_n\_per\_group**

Fast Channel num channels per group, default=1.

Type: `uint32_t`

Default value: 1

### **fast\_ch\_num\_groups**

Fast Channel num of groups, default=1.

Type: `uint32_t`

Default value: 1

### **`fast_ch_word_size`**

Fast Channel word size 32bit or 64bit, default=32.

Type: `uint32_t`

Default value: 32

### **`host_to_local.NUM_DB_CH`**

Number of doorbell channels.

Type: `uint32_t`

Default value: 1

### **`host_to_local.NUM_FAST_CH`**

Number of Fast Channels.

Type: `uint32_t`

Default value: 1

### **`host_to_local.NUM_FIFO_CH`**

Number of FIFO Channels.

Type: `uint32_t`

Default value: 1

### **`host_to_local.auto_op_full`**

AutoOp mode - AutoOp(min) == false, AutoOp(full) == true - default: AutoOp(min).

Type: `bool`

Default value: `false`

### **`host_to_local.diagnostics`**

Diagnostics 0:FATAL 1:ERROR 2:WARNING 3:INFO 4:DEBUG, Default:WARNING(2).

Type: `uint8_t`

Default value: 2

### **`host_to_local.fast_ch_group_int_enable`**

Fast Channel group interrupts enable, default=false.



Type: `bool`

Default value: `false`

### **`host_to_local.fast_ch_n_per_group`**

Fast Channel num channels per group, default=1.

Type: `uint32_t`

Default value: 1

### **`host_to_local.fast_ch_num_groups`**

Fast Channel num of groups, default=1.

Type: `uint32_t`

Default value: 1

### **`host_to_local.fast_ch_word_size`**

Fast Channel word size 32bit or 64bit, default=32.

Type: `uint32_t`

Default value: 32

### **`host_to_local.fifo_depth`**

Depth of the FIFO = `fifo_depth` + 1.

Type: `uint16_t`

Default value: 4

### **`host_to_local.fmu_location`**

FMU LOCATION: 0-2 (0:SENDER 1:RECEIVER 2:BOTH).

Type: `uint32_t`

Default value: 0

### **`host_to_local.m16ba_spt`**

Mailbox 16 bit access support to FIFO registers.

Type: `"bool"`

Default value: 0

### **`host_to_local.m32ba_spt`**

Mailbox 32 bit access support to FIFO registers.

Type: "bool"

Default value: 1

#### **host\_to\_local.m64ba\_spt**

Mailbox 64 bit access support to FIFO registers.

Type: "bool"

Default value: 0

#### **host\_to\_local.m8ba\_spt**

Mailbox 8 bit access support to FIFO registers.

Type: "bool"

Default value: 0

#### **host\_to\_local.monolithic**

Monolithic or Distributed MHU - default: monolithic(true).

Type: bool

Default value: true

#### **host\_to\_local.p16ba\_spt**

Postbox 16 bit access support to FIFO registers.

Type: "bool"

Default value: 0

#### **host\_to\_local.p32ba\_spt**

Postbox 32 bit access support to FIFO registers.

Type: "bool"

Default value: 1

#### **host\_to\_local.p64ba\_spt**

Postbox 64 bit access support to FIFO registers.

Type: "bool"

Default value: 0

#### **host\_to\_local.p8ba\_spt**

Postbox 8 bit access support to FIFO registers.

Type: `"bool"`

Default value: `0`

#### **host\_to\_local\_rcv\_log.trace\_debug**

Enable tracing of debug transactions.

Type: `bool`

Default value: `false`

#### **host\_to\_local\_rcv\_log.trace\_snoops**

Enable tracing of ACE snoop requests.

Type: `bool`

Default value: `false`

#### **host\_to\_local\_snd\_log.trace\_debug**

Enable tracing of debug transactions.

Type: `bool`

Default value: `false`

#### **host\_to\_local\_snd\_log.trace\_snoops**

Enable tracing of ACE snoop requests.

Type: `bool`

Default value: `false`

#### **irq\_rcv\_combined\_host\_log.forward\_signal**

If true, trace signal and forward signal from 'in' to 'out'. If false, trace signal only without driving 'out' port.

Type: `bool`

Default value: `true`

#### **irq\_rcv\_combined\_local\_log.forward\_signal**

If true, trace signal and forward signal from 'in' to 'out'. If false, trace signal only without driving 'out' port.

Type: `bool`

Default value: `true`

**irq\_snd\_combined\_host\_log.forward\_signal**

If true, trace signal and forward signal from 'in' to 'out'. If false, trace signal only without driving 'out' port.

Type: `bool`

Default value: `true`

**irq\_snd\_combined\_local\_log.forward\_signal**

If true, trace signal and forward signal from 'in' to 'out'. If false, trace signal only without driving 'out' port.

Type: `bool`

Default value: `true`

**local\_to\_host.NUM\_DB\_CH**

Number of doorbell channels.

Type: `uint32_t`

Default value: `1`

**local\_to\_host.NUM\_FAST\_CH**

Number of Fast Channels.

Type: `uint32_t`

Default value: `1`

**local\_to\_host.NUM\_FIFO\_CH**

Number of FIFO Channels.

Type: `uint32_t`

Default value: `1`

**local\_to\_host.auto\_op\_full**

AutoOp mode - AutoOp(min) == false, AutoOp(full) == true - default: AutoOp(min).

Type: `bool`

Default value: `false`

**local\_to\_host.diagnostics**

Diagnostics 0:FATAL 1:ERROR 2:WARNING 3:INFO 4:DEBUG, Default:WARNING(2).

Type: `uint8_t`

Default value: 2

### **`local_to_host.fast_ch_group_int_enable`**

Fast Channel group interrupts enable, default=false.

Type: `bool`

Default value: `false`

### **`local_to_host.fast_ch_n_per_group`**

Fast Channel num channels per group, default=1.

Type: `uint32_t`

Default value: 1

### **`local_to_host.fast_ch_num_groups`**

Fast Channel num of groups, default=1.

Type: `uint32_t`

Default value: 1

### **`local_to_host.fast_ch_word_size`**

Fast Channel word size 32bit or 64bit, default=32.

Type: `uint32_t`

Default value: 32

### **`local_to_host.fifo_depth`**

Depth of the FIFO = `fifo_depth` + 1.

Type: `uint16_t`

Default value: 4

### **`local_to_host.fmu_location`**

FMU LOCATION: 0-2 (0:SENDER 1:RECEIVER 2:BOTH).

Type: `uint32_t`

Default value: 0

### **`local_to_host.m16ba_spt`**

Mailbox 16 bit access support to FIFO registers.

Type: "bool"

Default value: 0

#### **local\_to\_host.m32ba\_spt**

Mailbox 32 bit access support to FIFO registers.

Type: "bool"

Default value: 1

#### **local\_to\_host.m64ba\_spt**

Mailbox 64 bit access support to FIFO registers.

Type: "bool"

Default value: 0

#### **local\_to\_host.m8ba\_spt**

Mailbox 8 bit access support to FIFO registers.

Type: "bool"

Default value: 0

#### **local\_to\_host.monolithic**

Monolithic or Distributed MHU - default: monolithic(true).

Type: bool

Default value: true

#### **local\_to\_host.p16ba\_spt**

Postbox 16 bit access support to FIFO registers.

Type: "bool"

Default value: 0

#### **local\_to\_host.p32ba\_spt**

Postbox 32 bit access support to FIFO registers.

Type: "bool"

Default value: 1

#### **local\_to\_host.p64ba\_spt**

Postbox 64 bit access support to FIFO registers.

Type: "bool"

Default value: 0

#### **local\_to\_host.p8ba\_spt**

Postbox 8 bit access support to FIFO registers.

Type: "bool"

Default value: 0

#### **local\_to\_host\_rcv\_log.trace\_debug**

Enable tracing of debug transactions.

Type: bool

Default value: false

#### **local\_to\_host\_rcv\_log.trace\_snoops**

Enable tracing of ACE snoop requests.

Type: bool

Default value: false

#### **local\_to\_host\_snd\_log.trace\_debug**

Enable tracing of debug transactions.

Type: bool

Default value: false

#### **local\_to\_host\_snd\_log.trace\_snoops**

Enable tracing of ACE snoop requests.

Type: bool

Default value: false

## 3.161 CoprocBus2SystemC

Defined in `examples/SystemCExport/Bridges/CoprocBus2SystemC.lisa`.

### About CoprocBus2SystemC

CoprocBusProtocol to SystemCCoprocBusProtocol converter.

## Iris and MTI instances for CoprocBus2SystemC

This model has the following Iris instances:

Name	Instance type
CoprocBus2SystemC	CoprocBus2SystemC

## Ports for CoprocBus2SystemC

Port	Direction	Protocol	Description
coproc_bus_s	slave	CoprocBusProtocol	-
sc_coproc_bus_m	master	SystemCCoprocBusProtocol	-

## Parameters for CoprocBus2SystemC

This component does not have any parameters.

# 3.162 CounterInterface2SystemC

Defined in `examples/SystemCExport/Bridges/CounterInterface2SystemC.lisa`.

## About CounterInterface2SystemC

CounterInterface to SystemC Converter.

## Iris and MTI instances for CounterInterface2SystemC

This model has the following Iris instances:

Name	Instance type
CounterInterface2SystemC	CounterInterface2SystemC

## Ports for CounterInterface2SystemC

Port	Direction	Protocol	Description
amba_pv_eventUpdate_m	master	AMBAPVValue	-
amba_pv_getCounterValue_s	slave	AMBAPVValueState64	-
amba_pv_requestEventUpdate_s	slave	AMBAPVValue64	-
amba_pv_requestSignalUpdate_s	slave	AMBAPVValue64	-
amba_pv_setEnabled_m	master	AMBAPVValue	-
amba_pv_signalUpdate_m	master	AMBAPVValue	-
cntvalueb	slave	CounterInterface	-

## Parameters for CounterInterface2SystemC

This component does not have any parameters.



## 3.163 D71

Defined in `LISA/D71.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following parameters were removed:

- `supports-v2`

### About D71

The model has the following limitations:

- No support for trusted layers.
- No support for image enhancements.
- No coprocessor support for HDR processing.
- No QoS support.
- The following configuration parameters are not available:
  - `CONFIG_MAX_LINE_SIZE`
  - `CONFIG_DISPLAY_TBU_EN`. TBUs are integrated separately using the given ports.
  - `CONFIG_AFBC_DMA_EN`. The ADU is present. If it is not used, do not program it.
- Some image formats are unsupported. For details, see the next section, Supported image formats.

### Supported image formats

#### ADU DS\_FORMAT

The `DS_FORMAT` register defines the image formats supported by the ADU DMA subsystem.

Support for the following image formats is implemented:

- All image formats supported.

#### ADU AES\_FORMAT

The `AES_FORMAT` register defines the image formats supported by the ADU AFBC encoding subsystem.

Support for the following image formats is implemented:

- `RGB_888`
- `RGBA_8888`

- YUV\_420\_P2\_8

## LS\_FORMAT/LR\_FORMAT

The `LS_FORMAT` and `LR_FORMAT` registers define the image formats supported by the main pipeline's layer processing unit.

Support for uncompressed images in the following formats is implemented:

- ARGB\_2101010
- BGRA\_1010102
- ARGB\_8888
- ABGR\_8888
- RGBA\_8888
- BGRA\_8888
- XRGB\_8888
- XBGR\_8888
- RGBX\_8888
- BGRX\_8888
- RGB\_888
- BGR\_888
- RGBA\_5551
- ABGR\_1555
- RGB\_565
- BGR\_565
- VYUY\_422\_P1\_8
- YVYU\_422\_P1\_8
- YUV\_420\_P2\_8
- YUV\_420\_P3\_8
- YUV\_420\_P1\_10
- YUV\_420\_P2\_10

Support for compressed images in the following formats is implemented:

- ABGR\_2101010
- ABGR\_8888
- BGR\_888
- ABGR\_1555
- BGR\_565
- YUV\_422\_P2\_8

- YUV\_420\_P2\_8
- YUV\_420\_P2\_10

## LW\_FORMAT

The `LW_FORMAT` register defines the image formats supported by the memory-writeback scheme performed by the layer processing unit.

Support for the following image formats is implemented:

- ARGB\_2101010
- ABGR\_2101010
- RGBA\_1010102
- BGRA\_1010102
- ARGB\_8888
- ABGR\_8888
- RGBA\_8888
- BGRA\_8888
- XRGB\_8888
- XBGR\_8888
- RGBX\_8888
- BGRX\_8888
- RGB\_888
- BGR\_888
- YUV\_420\_P2\_8

## Iris and MTI instances for D71

This model has the following Iris instances:

Name	Instance type
D71	D71
D71.apb_slave_adu	PVBusSlave
D71.apb_slave_dpu	PVBusSlave

This model has the following MTI trace components:

Name	Component type
D71	D71
D71.apb_slave_adu	PVBusSlave
D71.apb_slave_dpu	PVBusSlave

## Ports for D71

Port	Direction	Protocol	Description
apb_pvbus_s_adu	slave	PVBus	Slave port for register access.
apb_pvbus_s_dpu	slave	PVBus	-
axi_pvbus_adu_m	master	PVBus	Master AXI port for the AFBC unit
axi_pvbus_lpu_m	master	PVBus	Master AXI ports for pipelines
display_trace	master	FrameTracingProtocol	FrameTrace port
display	master	LCD	LCD ports for display outputs
irq0_gcu_out	master	Signal	Shared interrupt owned by the GCU
irq1_adu_out	master	Signal	Interrupt signal for the ADU block
pixelclock_in	slave	ClockSignal	Pixel clock inputs for the display outputs
pvbus_tbu_m	master	PVBus	Master ports for connection to TBU (SMMUv3)
pvbus_tbu_s	slave	PVBus	Slave ports for loopback from TBU (SMMUv3)
reset_signal	slave	Signal	Reset signal.

## Parameters for D71

### **adu\_nprot\_nsaid**

Non-protected NSAID for ADU transactions.

Type: `int`

Default value: 0

### **adu\_nprot\_s2\_sid**

Stage 2 non-protected StreamID for ADU transactions.

Type: `int`

Default value: 2

### **adu\_prot\_nsaid**

Protected NSAID for ADU transactions.

Type: `int`

Default value: 1

### **adu\_prot\_s2\_sid**

Stage 2 protected StreamID for ADU transactions.

Type: `int`

Default value: 5

**adu\_rd\_s1\_sid**

Stage 1 StreamID for ADU DMA read layer.

Type: `int`

Default value: 10

**adu\_wr\_s1\_sid**

Stage 1 StreamID for ADu AES write-back layer.

Type: `int`

Default value: 11

**display\_split\_en**

Display split enabled or not.

Type: `int`

Default value: 0

**force\_frame\_rate\_0**

If 0, PXLCLK0 is used. If >0 the model refreshes display output 0 at the rate per simulated second.

Type: `int`

Default value: 0

**force\_frame\_rate\_1**

If 0, PXLCLK1 is used. If >0 the model refreshes display output 1 at the rate per simulated second.

Type: `int`

Default value: 0

**lpu0\_10\_s1\_sid**

Stage 1 StreamID for LPU0 read layer 0.

Type: `int`

Default value: 0

**lpu0\_11\_s1\_sid**

Stage 1 StreamID for LPU0 read layer 1.

Type: `int`

Default value: 1

**`lpu0_l2_s1_sid`**

Stage 1 StreamID for LPU0 read layer 2.

Type: `int`

Default value: 2

**`lpu0_l3_s1_sid`**

Stage 1 StreamID for LPU0 read layer 3.

Type: `int`

Default value: 3

**`lpu0_nprot_nsaaid`**

Non-protected NSAID for LPU0 transactions.

Type: `int`

Default value: 0

**`lpu0_nprot_s2_sid`**

Stage 2 non-protected StreamID for LPU0 transactions.

Type: `int`

Default value: 0

**`lpu0_prot_nsaaid`**

Protected NSAID for LPU0 transactions.

Type: `int`

Default value: 1

**`lpu0_prot_s2_sid`**

Stage 2 protected StreamID for LPU0 transactions.

Type: `int`

Default value: 3

**`lpu0_wr_s1_sid`**

Stage 1 StreamID for LPU0 write-back layer.

Type: `int`

Default value: 8

**lpul\_10\_s1\_sid**

Stage 1 StreamID for LPU1 read layer 0.

Type: `int`

Default value: 4

**lpul\_11\_s1\_sid**

Stage 1 StreamID for LPU1 read layer 1.

Type: `int`

Default value: 5

**lpul\_12\_s1\_sid**

Stage 1 StreamID for LPU1 read layer 2.

Type: `int`

Default value: 6

**lpul\_13\_s1\_sid**

Stage 1 StreamID for LPU1 read layer 3.

Type: `int`

Default value: 7

**lpul\_nprot\_nsaid**

Non-protected NSAID for LPU1 transactions.

Type: `int`

Default value: 0

**lpul\_nprot\_s2\_sid**

Stage 2 non-protected StreamID for LPU1 transactions.

Type: `int`

Default value: 1

**lpul\_prot\_nsaid**

Protected NSAID for LPU1 transactions.

Type: `int`

Default value: 1

**lpul\_prot\_s2\_sid**

Stage 2 protected StreamID for LPU1 transactions.

Type: `int`

Default value: 4

**lpul\_wr\_s1\_sid**

Stage 1 StreamID for LPU1 write-back layer.

Type: `int`

Default value: 9

**num\_rich\_layers**

Number of Rich layers in each Layer Processing Unit.

Type: `int`

Default value: 2

### 3.164 DCSU

Defined in `LISA/DCSU.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

#### About DCSU

Diagnostic Control and Status Unit.

#### Iris and MTI instances for DCSU

This model has the following Iris instances:

Name	Instance type
DCSU	DCSU



## Ports for DCSU

Port	Direction	Protocol	Description
apb_pvbus_s	slave	PVBus	-
dcsu_irq_out	master	Signal	-
diag_dma_gpo_ch_in	slave	Value	-
diag_irq_mux_ctrl_in	slave	Signal	-
diag_lcm_seed_lfsr_valid_in	slave	Signal	-
diag_lcs_in	slave	Value	-
diag_lcs_valid_in	slave	Signal	-
diag_otpw_otp_is_ready_in	slave	Signal	-
diag_psi_dcu_en0_in	slave	Signal	-
diag_psi_dcu_en1_in	slave	Signal	-
diag_psi_gppc_in	slave	Signal	-
diag_psi_psi_status_in	slave	Signal	-
diag_psi_sam_in	slave	Signal	-
diag_trig_mux_ctrl_in	slave	Value	-
diag_trng_trigger_in	slave	Signal	-
die_id_out	master	Value	-
poreset	slave	Signal	-
post_code_out	master	Value	-

## Parameters for DCSU

### diagnostics

Diagnostics.

Type: uint32\_t

Default value: 2

### die\_id

die\_id.

Type: uint32\_t

Default value: 0

## 3.165 DMA350

Defined in `LISA/DMA350.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Preliminary support

For an explanation of the quality levels, see [Quality level definitions](#).

## About DMA350

This model supports the following functionality:

- 1-8 DMA channels
- 1D memory copy including increments, auto-reload, and command linking
- Interrupt capability for each channel
- 2D memory copy
- 1DWRAP and 2DWRAP support
- Template-based pack and unpack capability
- Security settings per channel
- Trigger input and output ports selectable for each channel
- General Purpose Output (GPO) per channel
- Streaming input and output interfaces per channel
- ADDR\_WIDTH and DATA\_WIDTH can be 32 bits or 64 bits

## Iris and MTI instances for DMA350

This model has the following Iris instances:

Name	Instance type
DMA350	<a href="#">DMA350</a>

## Ports for DMA350

Port	Direction	Protocol	Description
allch_pause_ack_nonsec	master	<a href="#">Signal</a>	-
allch_pause_ack_sec	master	<a href="#">Signal</a>	-
allch_pause_req_nonsec	slave	<a href="#">Signal</a>	Channel pause req/ack for all nonsecure channels
allch_pause_req_sec	slave	<a href="#">Signal</a>	Channel pause req/ack for all secure channels
allch_stop_ack_nonsec	master	<a href="#">Signal</a>	-
allch_stop_ack_sec	master	<a href="#">Signal</a>	-
allch_stop_req_nonsec	slave	<a href="#">Signal</a>	Channel stop req/ack for all nonsecure channels
allch_stop_req_sec	slave	<a href="#">Signal</a>	Channel stop req/ack for all secure channels
boot_addr	slave	<a href="#">Value_64</a>	Address when boot_en is enabled
boot_en	slave	<a href="#">Signal</a>	Enables channel 0 to load first command after reset from boot_addr
boot_memattr	slave	<a href="#">Value</a>	Memory attribute setting for the boot_addr
boot_shareattr	slave	<a href="#">Value</a>	Shareability attribute for the boot_attr
ch_enabled	master	<a href="#">Signal</a>	Enable status indicator per channel

Port	Direction	Protocol	Description
ch_err	master	Signal	Error status indicator per channel
ch_nonsec	master	Signal	Nonsecure status indicator per channel
ch_paused	master	Signal	Paused status indicator per channel
ch_priv	master	Signal	Privilege status indicator per channel
ch_stopped	master	Signal	Stopped status indicator per channel
clk_in	slave	ClockSignal	Ada DMA clock
gpo_ch	master	Value	MISC signals General purpose output for channels 0-15 Index refers to the channel
irq_channel	master	Signal	Channel IRQ Signals
irq_comb_nonsec_err	master	Signal	Nonsecure error IRQ Signal
irq_comb_nonsec	master	Signal	Nonsecure IRQ Signal
irq_comb_sec_err	master	Signal	Secure error IRQ Signal
irq_comb_sec	master	Signal	Secure IRQ Signal
irq_sec_viol_err	master	Signal	Security violation IRQ Signal
privileged_access_en_in	slave	Signal	Enables DMA privileged access generation during DMA_ICS sequence run
pvbus_m0	master	PVBus	AXI5 Master 0 Interface
pvbus_m1	master	PVBus	AXI5 Master 1 Interface
pvbus_s	slave	PVBus	APB4 Slave Interface
pvbus_stream_in	slave	PVBus	AXI-Stream In Interface
pvbus_stream_out	master	PVBus	AXI-Stream Out Interface
reset_in	slave	Signal	Ada DMA asynchronous reset
trig_in_ack_type	master	Value	-
trig_in_ack	master	Signal	Trigger In Acknowledgement Interface
trig_in_req_type	slave	Value	-
trig_in_req	slave	Signal	Trigger In Request Interface
trig_out_ack	slave	Signal	Trigger Out Acknowledgement Interface
trig_out_req	master	Signal	Trigger Out Request Interface

## Parameters for DMA350

### ADDR\_WIDTH

Address width of the bus interface.

Type: uint32\_t

Default value: 32

### AXI5\_M1\_ADDRESS\_RANGES

Address ranges for AXI5 M1 interface in the format e.g. [{"begin":0x40000000,"size":0x1000}, {"begin":0x80000000,"size":0x2000}]. Default when not specified uses AXI5 M0 interface.

Type: string

Default value: ""

**AXI5\_M1\_PRESENT**

Enables an additional master port.

Type: bool

Default value: 0

**CHID\_WIDTH**

Width of the configurable channel ID user signal. When set to 0, then the archid and awchid ports are not present on the module.

Type: uint8\_t

Default value: 0

**CH\_0\_FIFO\_DEPTH**

Sets the FIFO DEPTH the channels <N> can send for a transaction. N goes from 0 to NUM\_CHANNELS-1.

Type: uint8\_t

Default value: 2

**CH\_1\_FIFO\_DEPTH**

Sets the FIFO DEPTH the channels <N> can send for a transaction. N goes from 0 to NUM\_CHANNELS-1.

Type: uint8\_t

Default value: 2

**CH\_2\_FIFO\_DEPTH**

Sets the FIFO DEPTH the channels <N> can send for a transaction. N goes from 0 to NUM\_CHANNELS-1.

Type: uint8\_t

Default value: 2

**CH\_3\_FIFO\_DEPTH**

Sets the FIFO DEPTH the channels <N> can send for a transaction. N goes from 0 to NUM\_CHANNELS-1.

Type: uint8\_t

Default value: 2

**CH\_4\_FIFO\_DEPTH**

Sets the FIFO DEPTH the channels <N> can send for a transaction. N goes from 0 to NUM\_CHANNELS-1.

Type: `uint8_t`

Default value: 2

**CH\_5\_FIFO\_DEPTH**

Sets the FIFO DEPTH the channels <N> can send for a transaction. N goes from 0 to NUM\_CHANNELS-1.

Type: `uint8_t`

Default value: 2

**CH\_6\_FIFO\_DEPTH**

Sets the FIFO DEPTH the channels <N> can send for a transaction. N goes from 0 to NUM\_CHANNELS-1.

Type: `uint8_t`

Default value: 2

**CH\_7\_FIFO\_DEPTH**

Sets the FIFO DEPTH the channels <N> can send for a transaction. N goes from 0 to NUM\_CHANNELS-1.

Type: `uint8_t`

Default value: 2

**CH\_EXT\_FEAT\_EN**

Enabling the extended feature set for each channel. The extension contains 2D, WRAP, TMPLT features. Default value enables it for the number of channels.

Type: `bool`

Default value: `true`

**CH\_GPO\_EN**

Type: `bool`

Default value: `true`

**CH\_GPO\_MASK**

A bitmask for enabling the GPO port for each channel.

Type: `uint16_t`

Default value: `0`

### **CH\_STREAM\_EN**

Type: `bool`

Default value: `true`

### **CH\_STREAM\_MASK**

A bitmask for enabling the stream interfaces for each channel.

Type: `uint16_t`

Default value: `0`

### **DATA\_WIDTH**

Data width of the bus interface.

Type: `uint32_t`

Default value: `64`

### **DISABLE\_DEVICE**

Disable device and disable's all interfaces.

Type: `bool`

Default value: `false`

### **DUMP\_CONFIG**

Display DMA-350 DMAC parameters.

Type: `bool`

Default value: `false`

### **GPO\_WIDTH**

Width of GPO output for every channel.

Type: `uint8_t`

Default value: `1`

### **NUM\_CHANNELS**

Number of configurable DMA channels.

Type: `uint8_t`

Default value: 2

#### **NUM\_TRIGGER\_IN**

Number of trigger input ports.

Type: `uint8_t`

Default value: 2

#### **NUM\_TRIGGER\_OUT**

Number of trigger output ports.

Type: `uint8_t`

Default value: 2

#### **SECEXT\_PRESENT**

Enables Trustzone security support.

Type: `bool`

Default value: 1

## 3.166 DMC500

Defined in `LISA/DMC500.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### About DMC500

A platform can have multiple instances of this component. For example:

```
//LISA instantiation
composition
{
    // Memory controllers
    dmc0    : DMC500("default_region_attributes"=dmc_default_region_attributes,
                    "default_region_id_access"=dmc_default_region_id_access,
                    "passthrough_debug_access"=true);
    dmc1    : DMC500("default_region_attributes"=dmc_default_region_attributes,
                    "default_region_id_access"=dmc_default_region_id_access,
                    "passthrough_debug_access"=true);
}
```

## Differences between the model and the RTL

The model has the following limitations:

- It does not support address striping.
- It works with linear addresses and not in rank,bank,row,column form.
- It does not include any mechanism for error injection or detection.
- Scrubbing functionality is only provided from the interface point of view.
- It does not implement direct read or write commands.
- It does not implement any performance counters.
- All OR'd interrupt signals are missing from this release of the model. Users can connect the failed access interrupt as a substitute.
- The model combines separate failed access interrupts for system interfaces 1 and 2 into a single failed access interrupt.
- DMC-500 has three separate reset signals whereas this model has a single reset signal which supports the combined assertion of three resets. This model does not support separate reset signals.

## Iris and MTI instances for DMC500

This model has the following Iris instances:

Name	Instance type
DMC500	DMC500
DMC500.busslave	PVBusSlave

This model has the following MTI trace components:

Name	Component type
DMC500	DMC-500
DMC500.busslave	PVBusSlave

## Ports for DMC500

Port	Direction	Protocol	Description
apb_pvbus_s	slave	PVBus	Programmers interface to program and control the DMC-500.
failed_access_interrupt_signal	master	Signal	The DMC has detected a system request that has failed a permissions check and a previously detected assertion was not cleared.
filter_pvbus_m	master	PVBus	DMC master port from System Interface 0 to memory.
filter_pvbus_s	slave	PVBus	System interface 0. Generally, Non-coherent Interface.
reset_signal	slave	Signal	DMC reset.
si1_filter_pvbus_m	master	PVBus	DMC master port from System Interface 1 to memory.
si1_filter_pvbus_s	slave	PVBus	System interface 1. Generally, Coherent Interface.



## Parameters for DMC500

### **default\_region\_attributes**

Default Region Secure attributes. Only bits 31,30 set Secure RD/WR enable.

Type: `uint32_t`

Default value: `0x1`

### **default\_region\_id\_access**

Default Region NSAID permissions. Bits 31-16 set non-secure WR enable and bits 15-0 set non-secure RD enable.

Type: `uint32_t`

Default value: `0`

### **passthrough\_debug\_access**

Always allow debug access to memory.

Type: `bool`

Default value: `false`

## 3.167 DMC520

Defined in `LISA/DMC520.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r1p1	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### About DMC520

A platform can have multiple instances of this component. For example:

```
//LISA instantiation
composition
{
    // Memory controllers
    dmc520_0      : DMC520("passthrough_debug_access=true");
    dmc520_1      : DMC520("passthrough_debug_access=true");
}
```

## Limitations

- The model does not support address striping.
- It works with linear addresses and not in rank, bank, row, column form.
- It does not include any mechanism for error injection or detection.
- Scrubbing functionality is only provided from the interface point of view.
- It does not implement direct read or write commands.
- It does not implement any performance counters.

## Differences between the model and the RTL

The DMC520 and DMC620 models have different interfaces to those in the hardware due to the level of abstraction of memory in Fast Models. These are the differences:

- Like the hardware, the model has a slave port for configuring register accesses, `apb_pvbus_s`, and an AXI interface for incoming memory transactions that are attempting to access memory that is managed by the DMC.
- The hardware component translates incoming transactions on the AXI interface to a format that is conducive to accessing DRAM chips. The model performs TrustZone access control and models the DMC readiness state, but does not translate the transactions. If allowed, the model forwards incoming transactions to be handled by a memory storage handling component that works at the transaction level.

## Iris and MTI instances for DMC520

This model has the following Iris instances:

Name	Instance type
DMC520	DMC520
DMC520.busslave	PVBusSlave

This model has the following MTI trace components:

Name	Component type
DMC520	DMC-520
DMC520.busslave	PVBusSlave

## Ports for DMC520

Port	Direction	Protocol	Description
<code>all_or_interrupt_signal</code>	master	Signal	A combined interrupt that is the logical OR of the other interrupts.
<code>apb_pvbus_s</code>	slave	PVBus	Programmers interface to program and control the DMC-520.
<code>arch_fsm_interrupt_signal</code>	master	Signal	The DMC has detected a change in the architectural state.
<code>failed_access_interrupt_signal</code>	master	Signal	The DMC has detected a system request that has failed a permissions check and a previously detected assertion was not cleared.
<code>filter_pvbus_m</code>	master	PVBus	DMC master port to memory.

Port	Direction	Protocol	Description
filter_pvbus_s	slave	PVBus	System interface.
reset_signal	slave	Signal	DMC reset.
scrub_event_in	slave	Signal	Scrub event n trigger.
scrub_event_out	master	Signal	Scrub event n triggered.

## Parameters for DMC520

### **override\_default\_config**

Override default block-all behavior of DMC. Allow access to memory.

Type: `bool`

Default value: `false`

### **passthrough\_debug\_access**

Always allow debug access to memory.

Type: `bool`

Default value: `false`

## 3.168 DMC620

Defined in `LISA/DMC620.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r1p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Limitations

- It does not support address striping.
- It works with linear addresses and not in rank,bank,row,column form.
- It includes error injection and detection mechanisms and syndrome registers support only for RAS error types 4 (ECC single-bit SRAM error) and 5 (ECC double-bit SRAM error).
- Scrubbing functionality is not provided.
- It does not implement direct read or write commands.
- It does not implement any performance counters.

## Differences between the model and the RTL

The DMC520 and DMC620 models have different interfaces to those in the hardware due to the level of abstraction of memory in Fast Models. These are the differences:

- Like the hardware, the model has a slave port for configuring register accesses, `apb_pvbus_s`, and an AXI interface for incoming memory transactions that are attempting to access memory that is managed by the DMC.
- The hardware component translates incoming transactions on the AXI interface to a format that is conducive to accessing DRAM chips. The model performs TrustZone access control and models the DMC readiness state, but does not translate the transactions. If allowed, the model forwards incoming transactions to be handled by a memory storage handling component that works at the transaction level.

## Iris and MTI instances for DMC620

This model has the following Iris instances:

Name	Instance type
DMC620	DMC620
DMC620.busslave	PVBusSlave

This model has the following MTI trace components:

Name	Component type
DMC620	DMC-620
DMC620.busslave	PVBusSlave

## Ports for DMC620

Port	Direction	Protocol	Description
all_or_interrupt_signal	master	Signal	A combined interrupt that is the logical OR of the other interrupts.
apb_pvbus_s	slave	PVBus	Programmers interface to program and control the DMC-620.
arch_fsm_interrupt_signal	master	Signal	The DMC has detected a change in the architectural state.
failed_access_interrupt_signal	master	Signal	The DMC has detected a system request that has failed a permissions check and a previously detected assertion was not cleared.
filter_pvbus_m	master	PVBus	DMC master port to memory.
filter_pvbus_s	slave	PVBus	System interface.
interrupt_cfh_master	master	Signal	The DMC has detected and corrected a single bit error on the RAM access.
interrupt_combined_oflow_master	master	Signal	The DMC has detected a counter overflow.
interrupt_fh_master	master	Signal	The DMC has detected a double bit error on the RAM access.
reset_signal	slave	Signal	DMC reset.

## Parameters for DMC620

### **override\_default\_config**

Override default block-all behavior of DMC. Allow access to memory.

Type: `bool`

Default value: `false`

### **passthrough\_debug\_access**

Always allow debug access to memory.

Type: `bool`

Default value: `false`

## 3.169 DMC\_400

Defined in `LISA/DMC_400.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p1	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### About DMC\_400

The configuration of this model by setting the registers does not generally affect accesses to main memory.

This component has no timing information, so changing the values of the timing registers has no effect on behavior. The memory models do not attach to the component, and error checking does not update registers because the model does not include the possibility of errors.

### Iris and MTI instances for DMC\_400

This model has the following Iris instances:

Name	Instance type
DMC_400	DMC_400
DMC_400.apb_slave	PVBusSlave
DMC_400.ex_monY (where Y = 0–3)	PVBusMapper

This model has the following MTI trace components:

Name	Component type
DMC_400.apb_slave	PVBusSlave
DMC_400.ex_monY (where Y = 0-3)	PVBusMapper

## Ports for DMC\_400

Port	Direction	Protocol	Description
apb_interface	slave	PVBus	Slave bus interface for register access.
axi_if_in	slave	PVBus	Slave bus for connecting to bus decoder.
axi_if_out	master	PVBus	Master to connect to DRAM.
clr_ex_mon	master	Signal	Indicates when global monitors state is cleared.
user_status_ext	slave	Value	Allow user status to be set from outside.

## Parameters for DMC\_400

### ECC\_SUPPORT

Does the controller support ECC?.

Type: bool

Default value: true

### IF\_CHIP0

Set this parameter to 0 if memory is connected.

Type: int

Default value: -1

### IF\_CHIP1

Set this parameter to 0 if memory is connected.

Type: int

Default value: -1

### IF\_CHIP2

Set this parameter to 0 if memory is connected.

Type: int

Default value: -1

### IF\_CHIP3

Set this parameter to 0 if memory is connected.

Type: int

Default value: -1

### **MEMORY\_WIDTH**

Valid widths are 16, 32 or 64 bits.

Type: int

Default value: 32

### **diagnostics**

Diagnostics.

Type: uint32\_t

Default value: 0

### **revision\_string**

Revision.

Type: string

Default value: "r0p1"

## 3.170 DMS\_SUPER\_CSR

Defined in LISA/dms\_super\_csr.lisa.

### About DMS\_SUPER\_CSR

DMS Super Control Status Register.

### Iris and MTI instances for DMS\_SUPER\_CSR

This model has the following Iris instances:

Name	Instance type
DMS_SUPER_CSR	DMS_SUPER_CSR

### Ports for DMS\_SUPER\_CSR

Port	Direction	Protocol	Description
ap_interrupts_out	master	Signal	-
interrupts_in	slave	Signal	-
pvbuss	slave	PVBus	Interface to access dms_super_csr register
scp_interrupts_out	master	Signal	-

## Parameters for DMS\_SUPER\_CSR

### **diagnostics**

Diagnostics.

Type: `uint8_t`

Default value: 2

## 3.171 DTS

Defined in `LISA/DTS.lisa`.

### About DTS

Model can behave as DTSv2 or dDTS based on `model_behavior_mode` parameter configuration.

### Iris and MTI instances for DTS

This model has the following Iris instances:

Name	Instance type
DTS	<a href="#">DTS</a>

### Ports for DTS

Port	Direction	Protocol	Description
<code>dtb_temperature_in</code>	slave	<a href="#">ValueState</a>	temperature input value to get core temperature.
<code>pvtbus_s</code>	slave	<a href="#">PVBUS</a>	for mli and mgi data read

## Parameters for DTS

### **diagnostics**

Diagnostics.

Type: `uint8_t`

Default value: 2

### **model\_behavior\_mode**

Model should behave as DTSV2 or dDTS.

Type: `string`

Default value: "DTSV2"



**num\_of\_rsp\_connected**

Number of maximum RSPs can be connected to the DTSV2 or dDTS.

Type: uint8\_t

Default value: 9

## 3.172 DVFSM

Defined in `LISA/DVFSM.lisa`.

### About DVFSM

Dynamic Voltage and Frequency Scaling Manager.

### Iris and MTI instances for DVFSM

This model has the following Iris instances:

Name	Instance type
DVFSM	DVFSM
DVFSM.clock_muxX (where X = 0-1)	Clock_Multiplexer

### Ports for DVFSM

Port	Direction	Protocol	Description
ccsm_glcm_sel	master	Value	-
cfm_clkssel_cur	slave	ValueState	-
cfm_clkssel_override_ack	slave	Signal	-
cfm_clkssel_override_req	master	ValueState	-
cfm_clkssel_override_val	master	ValueState	-
cfm_mask_ack	slave	Signal	-
cfm_mask_req	master	Signal	-
clkkin_pll0	slave	ClockSignal	-
clkkin_pll1	slave	ClockSignal	-
clkout_fb	master	ClockSignal	-
clkout_nom	master	ClockSignal	-
dvfsm_ctrl_out	master	Signal	-
dvfsm_data_in	slave	ValueState	-
dvfsm_dynamic_out	master	ValueState	-
dvfsm_event_in	slave	Signal	-
dvfsm_irq	master	Signal	-
dvfsm_static_out	master	ValueState	-
pll0_bypass_en	master	Signal	-
pll0_dac_en	master	Signal	-

Port	Direction	Protocol	Description
pll0_dsm_en	master	Signal	-
pll0_dynamic_en	master	Signal	-
pll0_dynamic	master	ValueState	-
pll0_en	master	Signal	-
pll0_fbdiv	master	ValueState	-
pll0_fj_done	slave	Signal	-
pll0_fj_frac_accuracy	master	ValueState	-
pll0_fj_freq_ch_slope	master	ValueState	-
pll0_fj_freq_ch_tau	master	ValueState	-
pll0_fj_start	master	Signal	-
pll0_foutpostdiv_en	master	Signal	-
pll0_foutvco_en	master	Signal	-
pll0_glcm_sel	master	Value	-
pll0_lock	slave	Signal	-
pll0_offsetcal_en	master	Signal	-
pll0_offsetcalbyp	master	Signal	-
pll0_offsetcalcnt	master	ValueState	-
pll0_offsetcalin	master	ValueState	-
pll0_offsetfastcal	master	Signal	-
pll0_postdiv1	master	ValueState	-
pll0_postdiv2	master	ValueState	-
pll0_refdiv	master	ValueState	-
pll0_rsvd	master	ValueState	-
pll0_static	master	ValueState	-
pll0_status	slave	ValueState	-
pll1_bypass_en	master	Signal	-
pll1_dac_en	master	Signal	-
pll1_dsm_en	master	Signal	-
pll1_dynamic_en	master	Signal	-
pll1_dynamic	master	ValueState	-
pll1_en	master	Signal	-
pll1_fbdiv	master	ValueState	-
pll1_fj_done	slave	Signal	-
pll1_fj_frac_accuracy	master	ValueState	-
pll1_fj_freq_ch_slope	master	ValueState	-
pll1_fj_freq_ch_tau	master	ValueState	-
pll1_fj_start	master	Signal	-
pll1_foutpostdiv_en	master	Signal	-
pll1_foutvco_en	master	Signal	-
pll1_glcm_sel	master	Value	-

Port	Direction	Protocol	Description
pll1_lock	slave	Signal	-
pll1_offsetcal_en	master	Signal	-
pll1_offsetcalbyp	master	Signal	-
pll1_offsetcalcnt	master	ValueState	-
pll1_offsetcalin	master	ValueState	-
pll1_offsetfastcal	master	Signal	-
pll1_postdiv1	master	ValueState	-
pll1_postdiv2	master	ValueState	-
pll1_refdiv	master	ValueState	-
pll1_rsvd	master	ValueState	-
pll1_static	master	ValueState	-
pll1_status	slave	ValueState	-
PORESETn	slave	Signal	-
reg_pvbus_s	slave	PVBus	-
RESETn	slave	Signal	-
spare_in	slave	Signal	-
spare_out	master	Signal	-
sysclk_in	slave	ClockSignal	-
sysclk_qactive	master	Signal	-

## Parameters for DVFSM

### diagnostics

Diagnostics.

Type: uint8\_t

Default value: 2

### num\_dvfsm\_ctrl\_out

Number of bits in DVFSM\_CTRL\_OUT output.

Type: uint8\_t

Default value: 1

### num\_dvfsm\_data\_in

Number of DVFSM\_DATA\_IN inputs.

Type: uint8\_t

Default value: 1

**num\_dvfsm\_dynamic\_out**

Number of DVFSM\_DYNAMIC\_OUT outputs.

Type: uint8\_t

Default value: 7

**num\_dvfsm\_event\_in**

Number of bits in DVFSM\_EVENT\_IN inputs.

Type: uint8\_t

Default value: 3

**num\_dvfsm\_pse\_instr**

Maximum number of PSE instructions supported: 64 or 128.

Type: uint8\_t

Default value: 128

**num\_dvfsm\_static\_out**

Number of DVFSM\_STATIC\_OUT outputs.

Type: uint8\_t

Default value: 3

**num\_pll\_dynamic\_out**

Number of PLL Dynamic Setting outputs.

Type: uint8\_t

Default value: 3

**num\_pll\_static\_out**

Number of PLL Static Setting outputs.

Type: uint8\_t

Default value: 5

**num\_pll\_status\_in**

Number of PLL Status inputs.

Type: uint8\_t

Default value: 1

### 3.173 DebugAccessPort

Defined in `LISA/DebugAccessPort.lisa`.

#### Iris and MTI instances for DebugAccessPort

This model has the following Iris instances:

Name	Instance type
DebugAccessPort	dap

#### Ports for DebugAccessPort

Port	Direction	Protocol	Description
ap_pvbuss_m	master	PVBus	Debug access ports to bus master channels 0 and 1
clock	slave	ClockSignal	Clock input
paddrdbg31	master	Signal	Configurable output signal that indicates which master the access came from, AP0 or AP1

#### Parameters for DebugAccessPort

##### ap0\_has\_debug\_rom

Whether AP0 has a Debug ROM.

Type: `bool`

Default value: `false`

##### ap0\_rom\_base\_address

ROM base address for AP 0.

Type: `uint64_t`

Default value: `0x0`

##### ap0\_set\_paddrdbg31

Set paddrdbg31 signal during accesses on AP0.

Type: `bool`

Default value: `false`

##### ap1\_has\_debug\_rom

Whether AP1 has a Debug ROM.

Type: `bool`

Default value: `false`

**ap1\_rom\_base\_address**

ROM base address for AP 1.

Type: `uint64_t`

Default value: `0x0`

**ap1\_set\_paddrdbg31**

Set paddrdbg31 signal during accesses on AP1.

Type: `bool`

Default value: `false`

3.174 DebugROM

Defined in `LISA/DebugROM.lisa`.

About DebugRom

Debug ROM complying to an ADIV5-like interface

Iris and MTI instances for DebugROM

This model has the following Iris instances:

Name	Instance type
DebugROM	<code>debug_rom</code>

Ports for DebugROM

Port	Direction	Protocol	Description
<code>paddrdbg31</code>	master	Signal	Signal port for paddrdbg to recognize an external access
<code>pvbuss_s</code>	slave	PVBus	Bus slave port for accessing registers

Parameters for DebugROM

**ROMDEVID**

Value of Debug Rom Device Identification Register.

Type: `uint32_t`

Default value: `0x0`

**ROMPIDR**

Value of Debug Rom Peripheral Identification Register.

Type: uint64\_t

Default value: 0x4000bb000

**ROMPRIDR0**

Value of Debug ROM Power RequestID Register.

Type: uint32\_t

Default value: 0x1

**customer\_modified**

Type: uint32\_t

Default value: 0x0

**entry\_0**

Offset of component 0.

Type: uint32\_t

Default value: 0x0

**entry\_1**

Offset of component 1.

Type: uint32\_t

Default value: 0x0

**entry\_10**

Offset of component 10.

Type: uint32\_t

Default value: 0x0

**entry\_11**

Offset of component 11.

Type: uint32\_t

Default value: 0x0

**entry\_12**

Offset of component 12.

Type: uint32\_t

Default value: 0x0

**entry\_13**

Offset of component 13.

Type: uint32\_t

Default value: 0x0

**entry\_14**

Offset of component 14.

Type: uint32\_t

Default value: 0x0

**entry\_15**

Offset of component 15.

Type: uint32\_t

Default value: 0x0

**entry\_16**

Offset of component 16.

Type: uint32\_t

Default value: 0x0

**entry\_17**

Offset of component 17.

Type: uint32\_t

Default value: 0x0

**entry\_18**

Offset of component 18.

Type: uint32\_t

Default value: 0x0



**entry\_19**

Offset of component 19.

Type: uint32\_t

Default value: 0x0

**entry\_2**

Offset of component 2.

Type: uint32\_t

Default value: 0x0

**entry\_20**

Offset of component 20.

Type: uint32\_t

Default value: 0x0

**entry\_21**

Offset of component 21.

Type: uint32\_t

Default value: 0x0

**entry\_22**

Offset of component 22.

Type: uint32\_t

Default value: 0x0

**entry\_23**

Offset of component 23.

Type: uint32\_t

Default value: 0x0

**entry\_24**

Offset of component 24.

Type: uint32\_t

Default value: 0x0

**entry\_25**

Offset of component 25.

Type: uint32\_t

Default value: 0x0

**entry\_26**

Offset of component 26.

Type: uint32\_t

Default value: 0x0

**entry\_27**

Offset of component 27.

Type: uint32\_t

Default value: 0x0

**entry\_28**

Offset of component 28.

Type: uint32\_t

Default value: 0x0

**entry\_29**

Offset of component 29.

Type: uint32\_t

Default value: 0x0

**entry\_3**

Offset of component 3.

Type: uint32\_t

Default value: 0x0

**entry\_30**

Offset of component 30.

Type: uint32\_t

Default value: 0x0

**entry\_31**

Offset of component 31.

Type: uint32\_t

Default value: 0x0

**entry\_32**

Offset of component 32.

Type: uint32\_t

Default value: 0x0

**entry\_33**

Offset of component 33.

Type: uint32\_t

Default value: 0x0

**entry\_34**

Offset of component 34.

Type: uint32\_t

Default value: 0x0

**entry\_35**

Offset of component 35.

Type: uint32\_t

Default value: 0x0

**entry\_36**

Offset of component 36.

Type: uint32\_t

Default value: 0x0

**entry\_37**

Offset of component 37.

Type: uint32\_t

Default value: 0x0

**entry\_38**

Offset of component 38.

Type: uint32\_t

Default value: 0x0

**entry\_39**

Offset of component 39.

Type: uint32\_t

Default value: 0x0

**entry\_4**

Offset of component 4.

Type: uint32\_t

Default value: 0x0

**entry\_40**

Offset of component 40.

Type: uint32\_t

Default value: 0x0

**entry\_41**

Offset of component 41.

Type: uint32\_t

Default value: 0x0

**entry\_42**

Offset of component 42.

Type: uint32\_t

Default value: 0x0

**entry\_43**

Offset of component 43.

Type: uint32\_t

Default value: 0x0

**entry\_44**

Offset of component 44.

Type: uint32\_t

Default value: 0x0

**entry\_45**

Offset of component 45.

Type: uint32\_t

Default value: 0x0

**entry\_46**

Offset of component 46.

Type: uint32\_t

Default value: 0x0

**entry\_47**

Offset of component 47.

Type: uint32\_t

Default value: 0x0

**entry\_48**

Offset of component 48.

Type: uint32\_t

Default value: 0x0

**entry\_49**

Offset of component 49.

Type: uint32\_t

Default value: 0x0

**entry\_5**

Offset of component 5.

Type: uint32\_t

Default value: 0x0

**entry\_50**

Offset of component 50.

Type: uint32\_t

Default value: 0x0

**entry\_51**

Offset of component 51.

Type: uint32\_t

Default value: 0x0

**entry\_52**

Offset of component 52.

Type: uint32\_t

Default value: 0x0

**entry\_53**

Offset of component 53.

Type: uint32\_t

Default value: 0x0

**entry\_54**

Offset of component 54.

Type: uint32\_t

Default value: 0x0

**entry\_55**

Offset of component 55.

Type: uint32\_t

Default value: 0x0

**entry\_56**

Offset of component 56.

Type: uint32\_t

Default value: 0x0

**entry\_57**

Offset of component 57.

Type: uint32\_t

Default value: 0x0

**entry\_58**

Offset of component 58.

Type: uint32\_t

Default value: 0x0

**entry\_59**

Offset of component 59.

Type: uint32\_t

Default value: 0x0

**entry\_6**

Offset of component 6.

Type: uint32\_t

Default value: 0x0

**entry\_60**

Offset of component 60.

Type: uint32\_t

Default value: 0x0

**entry\_61**

Offset of component 61.

Type: uint32\_t

Default value: 0x0

**entry\_62**

Offset of component 62.

Type: uint32\_t

Default value: 0x0

**entry\_63**

Offset of component 63.

Type: uint32\_t

Default value: 0x0

**entry\_7**

Offset of component 7.

Type: uint32\_t

Default value: 0x0

**entry\_8**

Offset of component 8.

Type: uint32\_t

Default value: 0x0

**entry\_9**

Offset of component 9.

Type: uint32\_t

Default value: 0x0

**manufacturer\_revision\_number**

Type: uint32\_t

Default value: 0x0

**part\_number**

Type: uint32\_t

Default value: 0x0

**revision**

Type: uint32\_t

Default value: 0x0



## 3.175 DualClusterSystemConfigurationBlock

Defined in `examples/LISA/Common/LISA/DualClusterSystemConfigurationBlock.lisa`.

### About DualClusterSystemConfigurationBlock

Dual Cluster System Configuration Block.

### Iris and MTI instances for DualClusterSystemConfigurationBlock

This model has the following Iris instances:

Name	Instance type
DualClusterSystemConfigurationBlock	DualClusterSystemConfigurationBlock
DualClusterSystemConfigurationBlock.pvbusslave	PVBusSlave

This model has the following MTI trace components:

Name	Component type
DualClusterSystemConfigurationBlock	DualClusterSystemConfigurationBlock
DualClusterSystemConfigurationBlock.pvbusslave	PVBusSlave

### Ports for DualClusterSystemConfigurationBlock

Port	Direction	Protocol	Description
clk_in	slave	ClockSignal	-
cluster0_cfgend	master	Signal	-
cluster0_cfgte	master	Signal	-
cluster0_clusterid	master	Value	-
cluster0_corereset	master	Signal	-
cluster0_cpuporeset	master	Signal	-
cluster0_cxreset	master	Signal	-
cluster0_eventi	peer	Signal	-
cluster0_evento	peer	Signal	-
cluster0_iminlen	master	Signal	-
cluster0_l2reset	master	Signal	-
cluster0_standbywfi	slave	Signal	-
cluster0_vinithi	master	Signal	-
cluster1_cfgend	master	Signal	-
cluster1_clusterid	master	Value	-
cluster1_corereset	master	Signal	-
cluster1_cpuporeset	master	Signal	-
cluster1_eventi	peer	Signal	-
cluster1_evento	peer	Signal	-
cluster1_scureset	master	Signal	-
cluster1_standbywfi	slave	Signal	-

Port	Direction	Protocol	Description
cluster1_teinit	master	Signal	-
cluster1_vinithi	master	Signal	-
daughter_leds_state	master	ValueState	-
daughter_user_switches	master	ValueState	-
intgen	master	Signal	-
periphbase_32	master	Value	-
periphbase	master	Value_64	-
pvbus	slave	PVBus	-
system_reset	master	Signal	-
vgic_configuration_port	master	v7_VGIC_Configuration_Protocol	-

## Parameters for DualClusterSystemConfigurationBlock

### CFG\_ACTIVECLUSTER

Select which cluster will come out of reset coming out of power-on: bit[0] for primary cluster (Cortex-A15), bit[1] for secondary cluster (Cortex-A7). Value 0 is not allowed as it will hold both clusters in reset indefinitely!.

Type: uint32\_t

Default value: 1

### Cluster0IdOnPOReset

ClusterId for primary cluster (Cortex-A15) on power-on reset.

Type: uint32\_t

Default value: 0

### Cluster1IdOnPOReset

ClusterId for secondary cluster (Cortex-A7) on power-on reset.

Type: uint32\_t

Default value: 1

### DCSCB\_PERIPHBASE

PERIPHBASE.

Type: uint64\_t

Default value: 0x1e000000

### DCS\_AID

DCS\_AID is the Auxiliary ID Register.

Type: `uint32_t`

Default value: 0

### **DCS\_ID**

The value returned by the DCS\_ID register.

Type: `uint32_t`

Default value: 0x41120000

### **DCS\_ID\_BUILD\_NUMBER**

DCS\_ID build number.

Type: `uint32_t`

Default value: 1

### **DCS\_LEDS**

DCS\_LEDS represents eight LEDs on the board that form an 8-bit value that can be r/w from the Dual Cluster System Configuration Block.

Type: `uint32_t`

Default value: 0

### **DCS\_SW**

DCS\_SW represents eight switches on the board that form an 8-bit value that can be read from the Dual Cluster System Configuration Block.

Type: `uint32_t`

Default value: 0

### **FlipVGICWiringForCluster0AndCluster1**

Flip the VGIC wiring round for cluster0 and cluster1. With this false, then cpu0 of cluster0 is cpu interface 0 on the VGIC. If this is true then cpu0 of cluster1 becomes cpu interface 0 on the VGIC.

Type: `bool`

Default value: `false`

### **INTGEN\_INTS**

Number of custom IRQs controlled by interrupt generator is  $\text{INTGEN\_INTS} * 32 + 32$ .

Type: `uint32_t`

Default value: 3

**NumberOfCoresInCluster0**

The number of cores in the primary cluster.

Type: `uint32_t`

Default value: 0

**NumberOfCoresInCluster1**

The number of cores in the secondary cluster.

Type: `uint32_t`

Default value: 0

**ResetValueOfDaughterUserSwitches**

Reset value of the user switches on the daughterboard.

Type: `uint32_t`

Default value: 0

**stop\_on\_sequence\_id**

If non-zero the `sequence_id` of the SW trace mechanism on which to halt the simulator.

Type: `unsigned`

Default value: 0

## 3.176 DummyAPB

Defined in `LISA/DummyAPB.lisa`.

**About DummyAPB**

Use this dummy **RAZ/WI** APB device component to ensure that software does not receive aborts for accesses to devices that should be part of the system, but are not modeled.

For validation purposes it is useful to have dummy devices that are mostly **RAZ/WI** but return the correct value when you read ID registers. You can do that with this component in the following ways:

- Specify `periphid_24` for peripherals that follow the Arm pattern of having 12 ID registers at the top of an APB frame. For example:

```
periphid_24="04000000c2b00b000df005b1"
```

You also must set `periph_framesize` to 4 or 64, depending on whether the peripheral has its registers in a 4 KB or 64 KB frame.

- Give a space-separated list of offset:value pairs in the `periphid_generic` parameter to define read-only values from particular offsets. For example:

```
periphid_generic="000:02468ace 1fc:13579bdf"
```

The number of hex digits used to specify the address is used to define the width of the address mask used. For example, `bc:02468ace` returns `02468ace` at reads from any address ending `bc`.

- Give a space-separated list of offset:default-value pairs in the `ram_generic` parameter to construct RAM. That is, the register at the relevant offset returns the default-value, but if changed, it returns the value that it is changed to.

## Iris and MTI instances for DummyAPB

This model has the following Iris instances:

Name	Instance type
DummyAPB	DummyAPB
DummyAPB.pvbuslave	PVBusSlave

This model has the following MTI trace components:

Name	Component type
DummyAPB.pvbuslave	PVBusSlave

## Ports for DummyAPB

Port	Direction	Protocol	Description
pvbus_s	slave	PVBus	Bus slave interface.

## Parameters for DummyAPB

### **fail**

Abort all accesses.

Type: `bool`

Default value: `false`

### **failmsg**

String to print when 'fail'=true and access occurred.

Type: `string`

Default value: `""`

**periph\_framesize**

Size of frame (4/64, indicating if ID is at xFD0 or xFFD0).

Type: `int`

Default value: `-1`

**periphid\_24**

24 hex digits for the 12 bytes of peripheral ID.

Type: `string`

Default value: `""`

**periphid\_generic**

Set of space-separated offset:value pairs for dwords of ID.

Type: `string`

Default value: `""`

**ram\_generic**

Set of space-separated offset:default pairs for writable dwords.

Type: `string`

Default value: `""`

**warn\_once**

Warn once for the invalid read and write access.

Type: `bool`

Default value: `true`

## 3.177 ElfLoader

Defined in `LISA/ElfLoader.lisa`.

### Changes in 11.31.15

The following parameters were added:

- `use_virtual_addr`

## About ElfLoader

ElfLoader provides an alternative method of loading ELF files into the system. It can load files in either of the following formats, or in gzip-compressed versions of them:

- ELF
- Motorola S-Record

## Load file format

ElfLoader optionally uses a load file to load code and data in an ELF file to any address in the physical address space before the simulation starts.

Specify the ELF file to load using the `elf` parameter and the load file using the `loadfile` parameter. The load file contains an entry for each section in the ELF file. An entry has the following format:

```
<Section name>, <VA>, <PA>, <Offset>, <Size>, <PAS>, <MECID>
```

Where:

### Section name

The name of a segment of continuous code and data to be preloaded. It must match a segment name in the ELF file. It can only contain alphanumeric characters a - z, A - Z, 0 - 9, and underscores (\_).

### VA

The Virtual Address of the code and data in hexadecimal. This is the same as the address of the segment in the ELF file.

### PA

The Physical Address in hexadecimal to which the code and data will be preloaded in memory.

### Offset

Must be set to 0x0. Reserved for future use.

### Size

The size of the data to be preloaded at the Physical Address (PA) in hexadecimal. It is the sum of the sizes of the individual ELF sections belonging to the same segment. It should match the `memsz` field of the corresponding segment in the ELF file.

### PAS

The Physical Address Space to which the code and data will be preloaded. The possible values are:

#### S

Secure PAS

#### NS

Non-secure PAS

#### RL

Realm PAS

**RT**

Root PAS

**MECID**

Memory Encryption Context ID (optional). It can be specified for any PAS.

The following rules apply to the load file:

- The load file definition starts with the line:

```
** ELF_SECTION_RELOC_START **
```

and ends with the line:

```
** ELF_SECTION_RELOC_END **
```

- Lines within the load file definition that begin with # and all lines outside the definition are treated as comments.
- The hexadecimal values in the VA and PA fields must begin with 0x.
- Fields are separated by a comma and all whitespace characters are ignored.
- Blank lines within the load file definition are ignored.

**Example load file**

This example load file shows a segment of size 15 KB with a start address in VA space of 0x500000. It is preloaded to address 0x600000 in non-secure physical memory:

```
** ELF_SECTION_RELOC_START **
# Section relocation for Code Segment 1
PRELOAD TEST_1, 0x500000, 0x600000, 0x0, 0x3C00, NS
** ELF_SECTION_RELOC_END **
```

**ns\_copy, realm\_copy, and root\_copy parameters**

The boolean parameters `ns_copy`, `realm_copy`, and `root_copy` are alternatives to `1file`. If `1file` is also specified, it takes precedence over them.

If you enable any of these boolean parameters, the ELF file is loaded to both Secure PAS and to the PAS in the parameter name. So, for example, enabling all three parameters loads the ELF file to all four PASes.

If you do not specify `1file` and do not enable any of the `<PAS>_copy` parameters, the file is loaded to Secure PAS only.

Use `1file` instead of the `<PAS>_copy` parameters if you want control over:

- Where different segments in the ELF file are loaded to
- Whether to load the ELF file to Secure PAS
- The PA to load to and the MECID to use



## ElfLoader environment variables

Use the following environment variables to debug and configure ElfLoader. They are not intended for normal use, but can help analyze problems when loading ELF files and provide temporary workarounds.

Use the `OBJECT_LOADER_PARAMETERS*` environment variables to set generic or format-specific parameters. They can contain a list of parameter value assignments, separated by `:` or `;` characters. For example:

bash:

```
export  
OBJECT_LOADER_PARAMETERS_EXAMPLE="param1=1:otherParameter=string:thirdParam=0"
```

tcsh:

```
setenv OBJECT_LOADER_PARAMETERS_EXAMPLE param1=1:otherParameter=string:thirdParam=0
```

These values override values that are set in other ways, for example through the model parameters.

### **OBJECT\_LOADER\_VERBOSE**

Sets the verbosity of the ElfLoader debug messages. The value is either a decimal or hexadecimal number, and can encode any combination of the following flags:

#### **V\_GENERAL**

1

#### **V\_SYMBOL**

2

#### **V\_SOURCEREF**

4

#### **V\_STARTADDRESS**

8

#### **V\_RAWDATA**

16

#### **V\_DATAPACKETS**

32

#### **V\_RAWDATADUMP**

64

#### **V\_RAWLOADER**

128

#### **V\_SPECIFICOBJECTINFO**

256

**V\_WARN**

0x00010000

If the value of `OBJECT_LOADER_VERBOSE` is an empty string, the loader is set to maximum verbosity, with all flags active.

**OBJECT\_LOADER\_PARAMETERS**

Sets parameters for all supported loader formats. Currently, the only valid parameter is `putPacketChunks`. Set it to 1 to push the loaded data into the model in large chunks. For example:

bash:

```
export OBJECT_LOADER_PARAMETERS="putPacketChunks=1"
tcsh:
```

```
setenv OBJECT_LOADER_PARAMETERS putPacketChunks=1
```

**OBJECT\_LOADER\_PARAMETERS\_ELF**

Sets the following parameters:

**ignoreProgramHeader**

Set to 1 to use the sections instead of the program header table to load code and data. Default is 0.

**useVirtualAddresses**

Whether to use physical or virtual addresses of program headers. This parameter only affects program headers, not sections. Values:

**0**

Use physical addresses of program headers.

**1**

Use virtual addresses of program headers.

**2**

Use physical addresses of program headers but if they are all zero, use virtual addresses instead. This is the default.

**3**

Use virtual addresses of program headers but if they are all zero, use physical addresses instead.

**loadLocalFunctionsForFunctionInfo**

Set to 1 to load local symbols as functions into the `DebugInfoDB/FunctionInfo`. The default is 0.

**loadWeakFunctionsForFunctionInfo**

Set to 1 to load weak symbols as functions into the `DebugInfoDB/FunctionInfo`. The default is 0.

**localFunctionPrefix**

The value is a string which is prepended to each local function symbol before it is put into the DebugInfoDB/FunctionInfo. The default is "".

**weakFunctionPrefix**

The value is a string which is prepended to each weak function symbol before it is put into the DebugInfoDB/FunctionInfo. The default is "".

**putLineInfoIntoSimulator**

Set to 1 to transfer data from LineInfoDB to the simulator (source references). The default is 0.

**tryToLoadExtLineInfo**

This numeric value is the extended address shift value that is accepted. If the value is  $< 0$  (default), the `.debug_line_ext` section is ignored in all cases. If the value is  $\geq 0$ , the `.debug_line_ext` section is loaded if present and if the value of symbol `debug_line_ext_shift_value__` is equal to `tryToLoadExtLineInfo`, otherwise the `.debug_line` (or `.line`) section is loaded.

**loadDWARFDebugInfo**

Set to 1 to load DWARF debug information if possible. Set to 0 to ignore it. The default is 1.

**loadNOBITSRegions**

Possible values are:

**0**

Never load zero-data regions marked as `SHT_NOBITS` or the difference between `p_memsz - p_filesz` in program headers.

**1**

Always load zero-data regions marked as `SHT_NOBITS` or the difference between `p_memsz - p_filesz` in program headers. This is the default.

**2**

Do not load zero-data regions marked as `SHT_NOBITS` or the difference between `p_memsz - p_filesz` in program headers for Arm machine type, else load them.

**OBJECT\_LOADER\_PARAMETERS\_S\_RECORD**

Sets the parameters that control how section boundaries, which are not defined for S-record, are recognized. The default section name at the beginning of the file is `default`. All these parameters are boolean:

**newSectionOnS0**

S0 record introduces a new section whose name is either `s0_n`, where `n` is an increasing number starting at zero or the content of the S0 record, see parameter `sectionNameInS0`.

**sectionNameInS0**

If true, section names are taken from the S0 records. This parameter is only relevant if `newSectionOnS0==true`.

**newSectionOnS5**

S5 record introduces a new section whose name is `s5_n`, where `n` is an increasing number starting at zero.

**newSectionOnS789**

Sx record introduces a new section whose name is `sx_n`, where `n` is an increasing number starting at zero, and `x` is in the range 7-9.

**Iris and MTI instances for ElfLoader**

This model has the following Iris instances:

Name	Instance type
ElfLoader	ElfLoader
ElfLoader.pvbus_busmaster	PVBusMaster

This model has the following MTI trace components:

Name	Component type
ElfLoader.pvbus_busmaster	PVBusMaster

**Ports for ElfLoader**

Port	Direction	Protocol	Description
pvbus_m	master	PVBus	Master port for all memory accesses.
start_address	master	Value_64	Provides a value reflecting the entry point of the last ELF image to be loaded.

**Parameters for ElfLoader****elf**

ELF file.

Type: `string`

Default value: `N/A`

**impdef\_copy**

DEPRECATED: Use `realm_copy` or `root_copy` parameters. load ELF file to implementation defined memory spaces, if load file is not specified.

Type: `bool`

Default value: `false`

**lfile**

load file for large address mapping.

Type: `string`

Default value: `N/A`

**ns\_copy**

copy whole file to NS memory space.

Type: `bool`

Default value: `true`

**output\_attributes\_parameter\_of\_core**

Encoding of various attributes on the bus.

Type: `string`

Default value: `"ExtendedID[54:39]=MPAM_PARTID, ExtendedID[38]=MPAM_SP[0], ExtendedID[37]=MPAM_SP[1], UserFlags[31:16]=IMPDEF2"`

**realm\_copy**

load ELF file to REALM memory spaces, if load file is not specified.

Type: `bool`

Default value: `false`

**root\_copy**

load ELF file to ROOT memory spaces, if load file is not specified.

Type: `bool`

Default value: `false`

**use\_virtual\_addr**

Use virtual addresses of program headers in ELF file.

Type: `bool`

Default value: `false`

3.178 EthosU55

Defined in `LISA/EthosU55.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r2p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

## About EthosU55

- The EthosU55 model does not expose its registers through Iris.
- The `resetrn_in` signal is active-LOW.

## Iris and MTI instances for EthosU55

This model has the following Iris instances:

Name	Instance type
EthosU55	EthosU55
EthosU55.ExportTest.EthosU55[0].pvbusmaster	PVBusMaster
EthosU55.pvbusmasterY (where Y = 0-1)	PVBusMaster
EthosU55.pvbusslave	PVBusSlave

This model has the following MTI trace components:

Name	Component type
EthosU55	EthosU55
EthosU55.ExportTest.EthosU55[0].pvbusmaster	PVBusMaster
EthosU55.pvbusmasterY (where Y = 0-1)	PVBusMaster
EthosU55.pvbusslave	PVBusSlave

## Ports for EthosU55

Port	Direction	Protocol	Description
clk_in	slave	<a href="#">ClockSignal</a>	NPU clock signal
irq_out	master	<a href="#">Signal</a>	Sends interrupt requests to the external host application processor
popl_in	slave	<a href="#">Signal</a>	The Power-On-Reset Privilege Level (PORPL). LOW means User level. HIGH means Privileged level
posl_in	slave	<a href="#">Signal</a>	The Power-On-Reset Security Level (PORSL). LOW means Secure. HIGH means Non-secure.
pvbus_m0	master	<a href="#">PVBus</a>	Port 0 for NPU to access external memory
pvbus_m1	master	<a href="#">PVBus</a>	Port 1 for NPU to access external memory
pvbus_s	slave	<a href="#">PVBus</a>	Port to access NPU control registers
resetrn_in	slave	<a href="#">Signal</a>	NPU reset signal

## Parameters for EthosU55

### diagnostics

Enables extra information messages from the component about the NPU configuration and the PORPL/PORSL port values. Use any nonzero integer to enable all messages or 0 to disable them.

Type: `uint32_t`

Default value: 0

**extra\_args**

To activate fast processing mode, which significantly improves the performance of the model, set this parameter to "-fast". In fast mode, NPU performance counters are not representative of counters on real hardware. We recommend you do not use any other value without instructions from Arm Technical Support.

Type: `string`

Default value: ""

**num\_macs**

Number of 8x8 MACs performed per cycle (32, 64, 128, or 256).

Type: `uint32_t`

Default value: 128

## 3.179 EthosU65

Defined in `LISA/EthosU65.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p1	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### About EthosU65

- The EthosU65 model does not expose its registers through Iris.
- The `resetn_in` signal is active-LOW.

### Iris and MTI instances for EthosU65

This model has the following Iris instances:

Name	Instance type
EthosU65	EthosU65
EthosU65.ExportTest.EthosU65[0].pvbusmaster	PVBusMaster
EthosU65.pvbusmasterY (where Y = 0-1)	PVBusMaster
EthosU65.pvbusslave	PVBusSlave

This model has the following MTI trace components:

Name	Component type
EthosU65	EthosU65
EthosU65.ExportTest.EthosU65[0].pvbusmaster	PVBusMaster
EthosU65.pvbusmasterY (where Y = 0-1)	PVBusMaster
EthosU65.pvbusslave	PVBusSlave

## Ports for EthosU65

Port	Direction	Protocol	Description
clk_in	slave	ClockSignal	NPU clock signal
irq_out	master	Signal	Sends interrupt requests to the external host application processor
popl_in	slave	Signal	The Power-On-Reset Privilege Level (PORPL). LOW means User level. HIGH means Privileged level
posl_in	slave	Signal	The Power-On-Reset Security Level (PORSL). LOW means Secure. HIGH means Non-secure.
pvbus_m0	master	PVBus	Port 0 for NPU to access external memory
pvbus_m1	master	PVBus	Port 1 for NPU to access external memory
pvbus_s	slave	PVBus	Port to access NPU control registers
resetn_in	slave	Signal	NPU reset signal

## Parameters for EthosU65

### diagnostics

Enables extra information messages from the component about the NPU configuration and the PORPL/PORSL port values. Use any nonzero integer to enable all messages or 0 to disable them.

Type: `uint32_t`

Default value: 0

### extra\_args

To activate fast processing mode, which significantly improves the performance of the model, set this parameter to "-fast". In fast mode, NPU performance counters are not representative of counters on real hardware. We recommend you do not use any other value without instructions from Arm Technical Support.

Type: `string`

Default value: ""

### num\_macs

Number of 8x8 MACs performed per cycle (256 or 512).

Type: `uint32_t`

Default value: 256



## 3.180 EthosU85

Defined in `LISA/EthosU85.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### About EthosU85

- The EthosU85 model does not expose its registers through Iris.
- The `resetsn_in` signal is active-LOW.

### Iris and MTI instances for EthosU85

This model has the following Iris instances:

Name	Instance type
EthosU85	EthosU85
EthosU85.ExportTest.EthosU85[0].pvbusmaster	PVBusMaster
EthosU85.pvbusmasterY (where Y = 0-1)	PVBusMaster
EthosU85.pvbusslave	PVBusSlave

This model has the following MTI trace components:

Name	Component type
EthosU85	EthosU85
EthosU85.ExportTest.EthosU85[0].pvbusmaster	PVBusMaster
EthosU85.pvbusmasterY (where Y = 0-1)	PVBusMaster
EthosU85.pvbusslave	PVBusSlave

### Ports for EthosU85

Port	Direction	Protocol	Description
cfgextcap_in	slave	Value	The configuration of capabilities for DRAM AXI ports (32 bits). Sampled with soft and hard reset.
cfgexthash0_in	slave	Value_64	The configuration of hash function for selecting among EXT ports (40 bits). Used to set the hash for AXI DRAM ports 0 and 1 if they are present. Sampled with soft and hard reset.
cfgsramcap_in	slave	Value	The configuration of capabilities for SRAM AXI ports (32 bits). Sampled with soft and hard reset.
cfgsramhash0_in	slave	Value_64	The configuration of hash function for selecting among SRAM ports (40 bits). Used to set the hash for AXI SRAM ports 0 and 1. Sampled with soft and hard reset.
cfgsramhash1_in	slave	Value_64	The configuration of hash function for selecting among SRAM ports (40 bits). Used to set the hash for AXI SRAM ports 2 and 3 if they are present. Sampled with soft and hard reset.
clk_in	slave	ClockSignal	NPU clock signal

Port	Direction	Protocol	Description
irq_out	master	Signal	Sends interrupt requests to the external host application processor, level triggered when HIGH.
popl_in	slave	Signal	The Power-On-Reset Privilege Level (PORPL). LOW means User level. HIGH means Privileged level
posl_in	slave	Signal	The Power-On-Reset Security Level (PORSL). LOW means Secure. HIGH means Non-secure.
pvbush_m0	master	PVBus	Port 0 for NPU to access external memory
pvbush_m1	master	PVBus	Port 1 for NPU to access external memory
pvbush_s	slave	PVBus	Port to access NPU control registers
resetsn_in	slave	Signal	NPU reset signal (active-LOW)

## Parameters for EthosU85

### diagnostics

Enables extra information messages from the component about the NPU configuration and the PORPL/PORSL port values. Use any nonzero integer to enable all messages or 0 to disable them.

Type: `uint32_t`

Default value: 0

### extra\_args

To activate fast processing mode, which significantly improves the performance of the model, set this parameter to "-fast". In fast mode, NPU performance counters are not representative of counters on real hardware. We recommend you do not use any other value without instructions from Arm Technical Support.

Type: `string`

Default value: ""

### num\_macs

Number of 8x8 MACs performed per cycle (128, 256, 512, 1024, or 2048).

Type: `uint32_t`

Default value: 128

## 3.181 Firewall

Defined in `LISA/Firewall.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
rOp0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

## About Firewall

Firewall IP.

## Iris and MTI instances for Firewall

This model has the following Iris instances:

Name	Instance type
Firewall	Firewall
Firewall.BusLoggerX (where X = 0–31)	PVBusLogger
Firewall.BusLoggerX.mapper (where X = 0–31)	PVBusMapper
Firewall.BusMapperX (where X = 0–31)	PVBusMapper
Firewall.bus_slave	PVBusSlave

This model has the following MTI trace components:

Name	Component type
Firewall	Firewall
Firewall.BusLoggerX (where X = 0–31)	PVBusLogger
Firewall.BusLoggerX.mapper (where X = 0–31)	PVBusMapper
Firewall.BusMapperX (where X = 0–31)	PVBusMapper
Firewall.bus_slave	PVBusSlave

## Ports for Firewall

Port	Direction	Protocol	Description
irq_signal_tamper	master	Signal	-
irq_signal	master	Signal	-
lockdown	slave	Signal	-
pvbus_component_m	master	PVBus	-
pvbus_component_s	slave	PVBus	-
pvbus_program_iface	slave	PVBus	-
reset_signal	slave	Signal	-

## Parameters for Firewall

### ADDR\_WIDTH

ADDR\_WIDTH.

Type: string









Name	Instance type
FlashLoader	FlashLoader

### Ports for FlashLoader

Port	Direction	Protocol	Description
flash_device0	master	FlashLoaderPort	Used to program a flash device.
flash_device1	master	FlashLoaderPort	Used to program a flash device.
flash_device2	master	FlashLoaderPort	Used to program a flash device.
flash_device3	master	FlashLoaderPort	Used to program a flash device.
warm_reset	slave	Signal	Reset signal from external master.

### Parameters for FlashLoader

#### Diagnostics

Diagnostics.

Type: `uint32_t`

Default value: `0`

#### **enable\_raw\_format**

When enabled the flash file is not compressed.

Type: `bool`

Default value: `false`

#### **fname**

Filename (Default '(none)' means: Do not load any file. An empty string will cause a warning.).

Type: `string`

Default value: `"(none)"`

#### **fnameWrite**

FilenameWrite (Default '(none)' means: Do not save any file. An empty string will cause a warning.).

Type: `string`

Default value: `"(none)"`

#### **write\_flash\_after\_reset**

Force write back to `fnameWrite` on resets.

Type: `bool`



Default value: `false`

## 3.183 FrameTracingComponent

Defined in `LISA/FrameTracingComponent.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
rOp0	Alpha support

For an explanation of the quality levels, see [Quality level definitions](#).

### About FrameTracingComponent

You can add the `FrameTracingComponent` to a platform to intercept frame buffers on the way to the display, and perform an action based on the content of the frame buffer. This is expected to be used to test components in the display path, for example GPUs, ISPs, displays, and video.

Connect the `FrameTracingComponent` to an HDLCD or D71 component using its `frame_trace_s` port to receive copies of displayed frames through `FrameTracingProtocol`.

### Configuration

Configure the behavior of the `FrameTracingComponent` using a JSON input file. The configuration file defines a list of actions, and optionally a list of frames.

A frame is simply a string identifier and a path to a binary file containing the frame buffer content. Actions refer to frames by using the string identifier.

The `frames` array is only required if one or more actions refer to specific frame identifiers. If actions operate purely on delays (e.g., starting after N frames), you may omit the `frames` array.

An action inspects frames passed using `FrameTracingProtocol` and calls specific procedures based on the content of these frames.

Each action takes different arguments. The `FrameTracingComponent` supports the following actions:

**Table 3-628: FrameTracingComponent actions**

Action	Description
EXIT	End the simulation on matching a trigger frame.
TIME	Record the wall clock time before a trigger frame or between two frames.
RECORD	Save frames to files for offline processing.

You can define an arbitrary number of frames and actions.

The following JSON contains an example defining some frames, and one of each action:

```
{
  "frames" : [
    {
      "id" : "idstring1",
      "path" : "filepath1"
    },
    {
      "id" : "idstringN",
      "path" : "filepathN"
    },
    <...>
  ],
  "actions" : [
    {
      "action" : "EXIT",
      "frame" : "idstring1",
      "delay" : N
    },
    {
      "action" : "TIME",
      "id" : "actionid",
      "start_frame" : "idstring2",
      "stop_frame" : "idstring3",
    },
    {
      "action" : "RECORD",
      "start_frame" : "idstring2",
      "stop_frame" : "idstring4",
      "prefix" : "output_file_prefix",
      "start_delay" : N,
      "stop_delay" : M
    },
    <...>
  ]
}
```

There are 6 types of parameter used in the JSON configuration:

### Frame identifier

An arbitrary string that uniquely identifies a particular frame. In the `frames` array, the `id` parameter defines the identifier. Actions only refer to frame identifiers.

### Action type

The `action` parameter in the `actions` array. Must contain a valid action type. These parameters are case-insensitive.

### Action identifier

Identifier to distinguish multiple actions of the same type in the log file.

### File path

May be absolute or relative. Relative paths are relative to the directory containing the configuration file.

### Output file prefix

May include path components, but the directory must exist. Relative paths are relative to the directory containing the configuration file. A sequential index is appended to this string when saving frame buffers.

### Integer frame count

A delay applied to an aspect of an action by a number of frames. When not set, these default to 0, meaning no delay.

When the `ignore_consecutive_duplicates` parameter is true, a duplicate frame is considered to be the same as the previous frame and is not considered for any actions.

**Table 3-629: Action types and parameters**

Action	Parameter	Type	Description
EXIT	frame	Frame identifier	On matching <code>frame</code> , exit the simulation after <code>delay</code> frames.
	delay	Frame count	If <code>frame</code> is not specified, exit after <code>delay</code> frames from the start of the simulation.
TIME	start_frame	Frame identifier	Record the wall clock time between seeing <code>start_frame</code> and <code>stop_frame</code> .
	stop_frame	Frame identifier	If <code>start_frame</code> is not specified, record from the first input frame until seeing <code>stop_frame</code> .
	id	Action identifier	Recorded time will be printed in the log along with the string passed via <code>id</code> parameter.
RECORD	start_frame	Frame identifier	Record frames to files for offline processing.
	stop_frame	Frame identifier	The recording starts <code>start_delay</code> frames after seeing <code>start_frame</code> .
	prefix	Output file prefix	If <code>start_frame</code> is not specified, recording starts <code>start_delay</code> frames after the start of the simulation.
	start_delay	Frame count	The recording terminates on receiving <code>stop_delay</code> frames after seeing <code>stop_frame</code> .
	stop_delay	Frame count	If <code>stop_frame</code> is not specified, the recording terminates <code>stop_delay</code> frames after the start of the recording.
			If <code>start_delay</code> or <code>stop_delay</code> are not specified, they are assumed to be 0.  Frames are saved to files with the common prefix <code>prefix</code> , and an incrementing identifier suffix.

### Limitations

The `FrameTracingComponent` does not try to handle frame buffer formats. It assumes that width, height, and bits-per-pixel are sufficient to describe the size of the expected buffer, and compares the full content of the buffers.

### Iris and MTI instances for FrameTracingComponent

This model has the following Iris instances:

Name	Instance type
<code>FrameTracingComponent</code>	<code>FrameTracingComponent</code>

## Ports for FrameTracingComponent

Port	Direction	Protocol	Description
frame_trace_s	slave	FrameTracingProtocol	Connect to <code>frame_trace_m</code> , the manager FrameTracingProtocol port of the source component to receive frame buffers as they are produced.

## Parameters for FrameTracingComponent

### **config\_path**

Path to a JSON file containing a description of what the FrameTracingComponent should do.

Type: `string`

Default value: `""`

### **enabled**

Enables the frame tracing component. When disabled, the config file is not read, and the component is inactive.

Type: `bool`

Default value: `true`

### **ignore\_consecutive\_duplicates**

When `true` duplicate frames are considered to be the 'same' as the previous frame and are not considered for any actions. Set to `false` to consider each frame regardless of content. This relies on the source component to only send updated frames.

Type: `bool`

Default value: `true`

### **log\_file**

File that records what the FrameTracingComponent has done. If empty nothing is recorded.

Type: `string`

Default value: `""`

## 3.184 FrequencyProbe

Defined in `LISA/FrequencyProbe.lisa`.

### About FrequencyProbe

Clock Frequency observer.

## Iris and MTI instances for FrequencyProbe

This model has the following Iris instances:

Name	Instance type
FrequencyProbe	FrequencyProbe

## Ports for FrequencyProbe

Port	Direction	Protocol	Description
clk_in	slave	ClockSignal	-
freq_changed	master	ValueState	-

## Parameters for FrequencyProbe

### diagnostics

Diagnostics.

Type: uint32\_t

Default value: 0

## 3.185 GIC500

Defined in `LISA/GIC500.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r1p1	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### About GIC500

This is a single-component implementation of the GICv3 architecture with support for 256 cores. You can configure the model to support a maximum of 32 clusters with 8 cores per cluster. Use it with an Armv8-A core to deliver interrupts. It supports a single Interrupt Translation Service for message-based interrupts. It supports the architectural features, but does not support the implementation defined features.

To use the GIC500 component, you must configure some parameters. For example:

```
gic500: GIC500 (
    "num_clusters" = 2,
    "cpus_per_cluster_0" = 4,
    "cpus_per_cluster_1" = 4,
    "reg-base" = 0x2c200000,
    "SPI-count" = 256
```

);



Note

- To print to stderr the memory map of any GICv3 or later models that are included in the platform, set the `FASTSIM_GIC_MEMORY_MAP` environment variable to 1.
- For writes to `GITS_TRANSLATE64R`, the MSIRewriter component can be used to create the correct data transactions based on the device ID. For more details about `GITS_TRANSLATE64R`, see the [MSIRewriter](#) component.

## ManagerID, ExtendedID, and UserFlags

This table shows how this model encodes information in the `ManagerID`, `ExtendedID`, and `UserFlags` bus attributes:

**Table 3-635: ManagerID, ExtendedID, and UserFlags**

PVBus attribute	Bits used	Property encoded	Notes
ManagerID	63:0	Manager ID that invoked the register access	–
ExtendedID	–	–	Not used
UserFlags	–	–	Not used



Note

The `pvbus_m` port does not use `ManagerID`, `ExtendedID`, or `UserFlags`.

## Iris and MTI instances for GIC500

This model has the following Iris instances:

Name	Instance type
GIC500	GIC500
GIC500.ITS0	GICv3InterruptTranslationService
GIC500.ITS0.ExportTest.GIC500.ITS0.pvbus_m[0].pvbusmaster	PVBusMaster
GIC500.rd_0	GICv3RedistributorInternal
GIC500.rd_0_0	GICv3RedistributorInternal
GIC500.rd_0_0_0	GICv3RedistributorInternal
GIC500.rd_0_0_0_0	GICv3Redistributor
GIC500.rd_0_0_0_0.ExportTest.GIC500.rd_0_0_0_0.pvbus_m[0].pvbusmaster	PVBusMaster
GIC500.rd_t1	GICv3Distributor

This model has the following MTI trace components:

Name	Component type
GIC500.ITS0	GICv3InterruptTranslationService
GIC500.ITS0.ExportTest.GIC500.ITS0.pvbus_m[0].pvbusmaster	PVBusMaster

Name	Component type
GIC500.rd_0	GICv3RedistributorInternal
GIC500.rd_0_0	GICv3RedistributorInternal
GIC500.rd_0_0_0	GICv3RedistributorInternal
GIC500.rd_0_0_0_0	GICv3Redistributor
GIC500.rd_0_0_0_0.ExportTest.GIC500.rd_0_0_0_0.pvbus_m[0].pvbusmaster	PVBusMaster
GIC500.rd_t1	GICv3Distributor

## Ports for GIC500

Port	Direction	Protocol	Description
cfgsdisable	slave	Signal	Disable some SPLs signal.
cpu_active_0	slave	Signal	cpu_active pins of cluster 0.
cpu_active_10	slave	Signal	cpu_active pins of cluster 10.
cpu_active_11	slave	Signal	cpu_active pins of cluster 11.
cpu_active_12	slave	Signal	cpu_active pins of cluster 12.
cpu_active_13	slave	Signal	cpu_active pins of cluster 13.
cpu_active_14	slave	Signal	cpu_active pins of cluster 14.
cpu_active_15	slave	Signal	cpu_active pins of cluster 15.
cpu_active_16	slave	Signal	cpu_active pins of cluster 16.
cpu_active_17	slave	Signal	cpu_active pins of cluster 17.
cpu_active_18	slave	Signal	cpu_active pins of cluster 18.
cpu_active_19	slave	Signal	cpu_active pins of cluster 19.
cpu_active_1	slave	Signal	cpu_active pins of cluster 1.
cpu_active_20	slave	Signal	cpu_active pins of cluster 20.
cpu_active_21	slave	Signal	cpu_active pins of cluster 21.
cpu_active_22	slave	Signal	cpu_active pins of cluster 22.
cpu_active_23	slave	Signal	cpu_active pins of cluster 23.
cpu_active_24	slave	Signal	cpu_active pins of cluster 24.
cpu_active_25	slave	Signal	cpu_active pins of cluster 25.
cpu_active_26	slave	Signal	cpu_active pins of cluster 26.
cpu_active_27	slave	Signal	cpu_active pins of cluster 27.
cpu_active_28	slave	Signal	cpu_active pins of cluster 28.
cpu_active_29	slave	Signal	cpu_active pins of cluster 29.
cpu_active_2	slave	Signal	cpu_active pins of cluster 2.
cpu_active_30	slave	Signal	cpu_active pins of cluster 30.
cpu_active_31	slave	Signal	cpu_active pins of cluster 31.
cpu_active_3	slave	Signal	cpu_active pins of cluster 3.
cpu_active_4	slave	Signal	cpu_active pins of cluster 4.
cpu_active_5	slave	Signal	cpu_active pins of cluster 5.
cpu_active_6	slave	Signal	cpu_active pins of cluster 6.
cpu_active_7	slave	Signal	cpu_active pins of cluster 7.

Port	Direction	Protocol	Description
cpu_active_8	slave	Signal	cpu_active pins of cluster 8.
cpu_active_9	slave	Signal	cpu_active pins of cluster 9.
po_reset	slave	Signal	Power on reset.
ppi16_in_0	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 0.
ppi16_in_10	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 10.
ppi16_in_11	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 11.
ppi16_in_12	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 12.
ppi16_in_13	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 13.
ppi16_in_14	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 14.
ppi16_in_15	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 15.
ppi16_in_16	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 16.
ppi16_in_17	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 17.
ppi16_in_18	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 18.
ppi16_in_19	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 19.
ppi16_in_1	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 1.
ppi16_in_20	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 20.
ppi16_in_21	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 21.
ppi16_in_22	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 22.
ppi16_in_23	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 23.
ppi16_in_24	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 24.
ppi16_in_25	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 25.
ppi16_in_26	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 26.
ppi16_in_27	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 27.
ppi16_in_28	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 28.
ppi16_in_29	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 29.
ppi16_in_2	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 2.
ppi16_in_30	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 30.
ppi16_in_31	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 31.
ppi16_in_3	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 3.
ppi16_in_4	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 4.
ppi16_in_5	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 5.
ppi16_in_6	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 6.
ppi16_in_7	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 7.
ppi16_in_8	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 8.
ppi16_in_9	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 9.
ppi17_in_0	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 0.
ppi17_in_10	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 10.
ppi17_in_11	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 11.
ppi17_in_12	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 12.
ppi17_in_13	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 13.



Port	Direction	Protocol	Description
ppi17_in_14	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 14.
ppi17_in_15	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 15.
ppi17_in_16	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 16.
ppi17_in_17	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 17.
ppi17_in_18	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 18.
ppi17_in_19	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 19.
ppi17_in_1	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 1.
ppi17_in_20	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 20.
ppi17_in_21	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 21.
ppi17_in_22	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 22.
ppi17_in_23	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 23.
ppi17_in_24	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 24.
ppi17_in_25	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 25.
ppi17_in_26	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 26.
ppi17_in_27	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 27.
ppi17_in_28	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 28.
ppi17_in_29	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 29.
ppi17_in_2	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 2.
ppi17_in_30	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 30.
ppi17_in_31	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 31.
ppi17_in_3	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 3.
ppi17_in_4	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 4.
ppi17_in_5	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 5.
ppi17_in_6	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 6.
ppi17_in_7	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 7.
ppi17_in_8	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 8.
ppi17_in_9	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 9.
ppi18_in_0	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 0.
ppi18_in_10	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 10.
ppi18_in_11	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 11.
ppi18_in_12	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 12.
ppi18_in_13	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 13.
ppi18_in_14	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 14.
ppi18_in_15	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 15.
ppi18_in_16	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 16.
ppi18_in_17	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 17.
ppi18_in_18	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 18.
ppi18_in_19	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 19.
ppi18_in_1	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 1.
ppi18_in_20	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 20.

Port	Direction	Protocol	Description
ppi18_in_21	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 21.
ppi18_in_22	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 22.
ppi18_in_23	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 23.
ppi18_in_24	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 24.
ppi18_in_25	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 25.
ppi18_in_26	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 26.
ppi18_in_27	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 27.
ppi18_in_28	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 28.
ppi18_in_29	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 29.
ppi18_in_2	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 2.
ppi18_in_30	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 30.
ppi18_in_31	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 31.
ppi18_in_3	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 3.
ppi18_in_4	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 4.
ppi18_in_5	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 5.
ppi18_in_6	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 6.
ppi18_in_7	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 7.
ppi18_in_8	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 8.
ppi18_in_9	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 9.
ppi19_in_0	slave	Signal	Private peripheral interrupts (ID19) of cluster 0.
ppi19_in_10	slave	Signal	Private peripheral interrupts (ID19) of cluster 10.
ppi19_in_11	slave	Signal	Private peripheral interrupts (ID19) of cluster 11.
ppi19_in_12	slave	Signal	Private peripheral interrupts (ID19) of cluster 12.
ppi19_in_13	slave	Signal	Private peripheral interrupts (ID19) of cluster 13.
ppi19_in_14	slave	Signal	Private peripheral interrupts (ID19) of cluster 14.
ppi19_in_15	slave	Signal	Private peripheral interrupts (ID19) of cluster 15.
ppi19_in_16	slave	Signal	Private peripheral interrupts (ID19) of cluster 16.
ppi19_in_17	slave	Signal	Private peripheral interrupts (ID19) of cluster 17.
ppi19_in_18	slave	Signal	Private peripheral interrupts (ID19) of cluster 18.
ppi19_in_19	slave	Signal	Private peripheral interrupts (ID19) of cluster 19.
ppi19_in_1	slave	Signal	Private peripheral interrupts (ID19) of cluster 1.
ppi19_in_20	slave	Signal	Private peripheral interrupts (ID19) of cluster 20.
ppi19_in_21	slave	Signal	Private peripheral interrupts (ID19) of cluster 21.
ppi19_in_22	slave	Signal	Private peripheral interrupts (ID19) of cluster 22.
ppi19_in_23	slave	Signal	Private peripheral interrupts (ID19) of cluster 23.
ppi19_in_24	slave	Signal	Private peripheral interrupts (ID19) of cluster 24.
ppi19_in_25	slave	Signal	Private peripheral interrupts (ID19) of cluster 25.
ppi19_in_26	slave	Signal	Private peripheral interrupts (ID19) of cluster 26.
ppi19_in_27	slave	Signal	Private peripheral interrupts (ID19) of cluster 27.
ppi19_in_28	slave	Signal	Private peripheral interrupts (ID19) of cluster 28.

Port	Direction	Protocol	Description
ppi19_in_29	slave	Signal	Private peripheral interrupts (ID19) of cluster 29.
ppi19_in_2	slave	Signal	Private peripheral interrupts (ID19) of cluster 2.
ppi19_in_30	slave	Signal	Private peripheral interrupts (ID19) of cluster 30.
ppi19_in_31	slave	Signal	Private peripheral interrupts (ID19) of cluster 31.
ppi19_in_3	slave	Signal	Private peripheral interrupts (ID19) of cluster 3.
ppi19_in_4	slave	Signal	Private peripheral interrupts (ID19) of cluster 4.
ppi19_in_5	slave	Signal	Private peripheral interrupts (ID19) of cluster 5.
ppi19_in_6	slave	Signal	Private peripheral interrupts (ID19) of cluster 6.
ppi19_in_7	slave	Signal	Private peripheral interrupts (ID19) of cluster 7.
ppi19_in_8	slave	Signal	Private peripheral interrupts (ID19) of cluster 8.
ppi19_in_9	slave	Signal	Private peripheral interrupts (ID19) of cluster 9.
ppi20_in_0	slave	Signal	Private peripheral interrupts (ID20) of cluster 0.
ppi20_in_10	slave	Signal	Private peripheral interrupts (ID20) of cluster 10.
ppi20_in_11	slave	Signal	Private peripheral interrupts (ID20) of cluster 11.
ppi20_in_12	slave	Signal	Private peripheral interrupts (ID20) of cluster 12.
ppi20_in_13	slave	Signal	Private peripheral interrupts (ID20) of cluster 13.
ppi20_in_14	slave	Signal	Private peripheral interrupts (ID20) of cluster 14.
ppi20_in_15	slave	Signal	Private peripheral interrupts (ID20) of cluster 15.
ppi20_in_16	slave	Signal	Private peripheral interrupts (ID20) of cluster 16.
ppi20_in_17	slave	Signal	Private peripheral interrupts (ID20) of cluster 17.
ppi20_in_18	slave	Signal	Private peripheral interrupts (ID20) of cluster 18.
ppi20_in_19	slave	Signal	Private peripheral interrupts (ID20) of cluster 19.
ppi20_in_1	slave	Signal	Private peripheral interrupts (ID20) of cluster 1.
ppi20_in_20	slave	Signal	Private peripheral interrupts (ID20) of cluster 20.
ppi20_in_21	slave	Signal	Private peripheral interrupts (ID20) of cluster 21.
ppi20_in_22	slave	Signal	Private peripheral interrupts (ID20) of cluster 22.
ppi20_in_23	slave	Signal	Private peripheral interrupts (ID20) of cluster 23.
ppi20_in_24	slave	Signal	Private peripheral interrupts (ID20) of cluster 24.
ppi20_in_25	slave	Signal	Private peripheral interrupts (ID20) of cluster 25.
ppi20_in_26	slave	Signal	Private peripheral interrupts (ID20) of cluster 26.
ppi20_in_27	slave	Signal	Private peripheral interrupts (ID20) of cluster 27.
ppi20_in_28	slave	Signal	Private peripheral interrupts (ID20) of cluster 28.
ppi20_in_29	slave	Signal	Private peripheral interrupts (ID20) of cluster 29.
ppi20_in_2	slave	Signal	Private peripheral interrupts (ID20) of cluster 2.
ppi20_in_30	slave	Signal	Private peripheral interrupts (ID20) of cluster 30.
ppi20_in_31	slave	Signal	Private peripheral interrupts (ID20) of cluster 31.
ppi20_in_3	slave	Signal	Private peripheral interrupts (ID20) of cluster 3.
ppi20_in_4	slave	Signal	Private peripheral interrupts (ID20) of cluster 4.
ppi20_in_5	slave	Signal	Private peripheral interrupts (ID20) of cluster 5.
ppi20_in_6	slave	Signal	Private peripheral interrupts (ID20) of cluster 6.

Port	Direction	Protocol	Description
ppi20_in_7	slave	Signal	Private peripheral interrupts (ID20) of cluster 7.
ppi20_in_8	slave	Signal	Private peripheral interrupts (ID20) of cluster 8.
ppi20_in_9	slave	Signal	Private peripheral interrupts (ID20) of cluster 9.
ppi21_in_0	slave	Signal	Private peripheral interrupts (ID21) of cluster 0.
ppi21_in_10	slave	Signal	Private peripheral interrupts (ID21) of cluster 10.
ppi21_in_11	slave	Signal	Private peripheral interrupts (ID21) of cluster 11.
ppi21_in_12	slave	Signal	Private peripheral interrupts (ID21) of cluster 12.
ppi21_in_13	slave	Signal	Private peripheral interrupts (ID21) of cluster 13.
ppi21_in_14	slave	Signal	Private peripheral interrupts (ID21) of cluster 14.
ppi21_in_15	slave	Signal	Private peripheral interrupts (ID21) of cluster 15.
ppi21_in_16	slave	Signal	Private peripheral interrupts (ID21) of cluster 16.
ppi21_in_17	slave	Signal	Private peripheral interrupts (ID21) of cluster 17.
ppi21_in_18	slave	Signal	Private peripheral interrupts (ID21) of cluster 18.
ppi21_in_19	slave	Signal	Private peripheral interrupts (ID21) of cluster 19.
ppi21_in_1	slave	Signal	Private peripheral interrupts (ID21) of cluster 1.
ppi21_in_20	slave	Signal	Private peripheral interrupts (ID21) of cluster 20.
ppi21_in_21	slave	Signal	Private peripheral interrupts (ID21) of cluster 21.
ppi21_in_22	slave	Signal	Private peripheral interrupts (ID21) of cluster 22.
ppi21_in_23	slave	Signal	Private peripheral interrupts (ID21) of cluster 23.
ppi21_in_24	slave	Signal	Private peripheral interrupts (ID21) of cluster 24.
ppi21_in_25	slave	Signal	Private peripheral interrupts (ID21) of cluster 25.
ppi21_in_26	slave	Signal	Private peripheral interrupts (ID21) of cluster 26.
ppi21_in_27	slave	Signal	Private peripheral interrupts (ID21) of cluster 27.
ppi21_in_28	slave	Signal	Private peripheral interrupts (ID21) of cluster 28.
ppi21_in_29	slave	Signal	Private peripheral interrupts (ID21) of cluster 29.
ppi21_in_2	slave	Signal	Private peripheral interrupts (ID21) of cluster 2.
ppi21_in_30	slave	Signal	Private peripheral interrupts (ID21) of cluster 30.
ppi21_in_31	slave	Signal	Private peripheral interrupts (ID21) of cluster 31.
ppi21_in_3	slave	Signal	Private peripheral interrupts (ID21) of cluster 3.
ppi21_in_4	slave	Signal	Private peripheral interrupts (ID21) of cluster 4.
ppi21_in_5	slave	Signal	Private peripheral interrupts (ID21) of cluster 5.
ppi21_in_6	slave	Signal	Private peripheral interrupts (ID21) of cluster 6.
ppi21_in_7	slave	Signal	Private peripheral interrupts (ID21) of cluster 7.
ppi21_in_8	slave	Signal	Private peripheral interrupts (ID21) of cluster 8.
ppi21_in_9	slave	Signal	Private peripheral interrupts (ID21) of cluster 9.
ppi22_in_0	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 0.
ppi22_in_10	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 10.
ppi22_in_11	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 11.

Port	Direction	Protocol	Description
ppi22_in_12	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 12.
ppi22_in_13	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 13.
ppi22_in_14	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 14.
ppi22_in_15	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 15.
ppi22_in_16	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 16.
ppi22_in_17	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 17.
ppi22_in_18	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 18.
ppi22_in_19	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 19.
ppi22_in_1	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 1.
ppi22_in_20	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 20.
ppi22_in_21	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 21.
ppi22_in_22	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 22.
ppi22_in_23	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 23.
ppi22_in_24	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 24.
ppi22_in_25	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 25.
ppi22_in_26	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 26.
ppi22_in_27	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 27.
ppi22_in_28	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 28.
ppi22_in_29	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 29.
ppi22_in_2	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 2.
ppi22_in_30	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 30.
ppi22_in_31	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 31.
ppi22_in_3	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 3.

Port	Direction	Protocol	Description
ppi22_in_4	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 4.
ppi22_in_5	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 5.
ppi22_in_6	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 6.
ppi22_in_7	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 7.
ppi22_in_8	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 8.
ppi22_in_9	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 9.
ppi23_in_0	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 0.
ppi23_in_10	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 10.
ppi23_in_11	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 11.
ppi23_in_12	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 12.
ppi23_in_13	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 13.
ppi23_in_14	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 14.
ppi23_in_15	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 15.
ppi23_in_16	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 16.
ppi23_in_17	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 17.
ppi23_in_18	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 18.
ppi23_in_19	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 19.
ppi23_in_1	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 1.
ppi23_in_20	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 20.
ppi23_in_21	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 21.
ppi23_in_22	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 22.
ppi23_in_23	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 23.
ppi23_in_24	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 24.

Port	Direction	Protocol	Description
ppi23_in_25	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 25.
ppi23_in_26	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 26.
ppi23_in_27	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 27.
ppi23_in_28	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 28.
ppi23_in_29	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 29.
ppi23_in_2	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 2.
ppi23_in_30	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 30.
ppi23_in_31	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 31.
ppi23_in_3	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 3.
ppi23_in_4	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 4.
ppi23_in_5	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 5.
ppi23_in_6	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 6.
ppi23_in_7	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 7.
ppi23_in_8	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 8.
ppi23_in_9	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 9.
ppi24_in_0	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 0.
ppi24_in_10	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 10.
ppi24_in_11	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 11.
ppi24_in_12	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 12.
ppi24_in_13	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 13.
ppi24_in_14	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 14.
ppi24_in_15	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 15.
ppi24_in_16	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 16.

Port	Direction	Protocol	Description
ppi24_in_17	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 17.
ppi24_in_18	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 18.
ppi24_in_19	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 19.
ppi24_in_1	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 1.
ppi24_in_20	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 20.
ppi24_in_21	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 21.
ppi24_in_22	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 22.
ppi24_in_23	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 23.
ppi24_in_24	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 24.
ppi24_in_25	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 25.
ppi24_in_26	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 26.
ppi24_in_27	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 27.
ppi24_in_28	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 28.
ppi24_in_29	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 29.
ppi24_in_2	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 2.
ppi24_in_30	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 30.
ppi24_in_31	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 31.
ppi24_in_3	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 3.
ppi24_in_4	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 4.
ppi24_in_5	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 5.
ppi24_in_6	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 6.
ppi24_in_7	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 7.
ppi24_in_8	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 8.



Port	Direction	Protocol	Description
ppi24_in_9	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 9.
ppi25_in_0	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 0.
ppi25_in_10	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 10.
ppi25_in_11	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 11.
ppi25_in_12	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 12.
ppi25_in_13	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 13.
ppi25_in_14	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 14.
ppi25_in_15	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 15.
ppi25_in_16	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 16.
ppi25_in_17	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 17.
ppi25_in_18	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 18.
ppi25_in_19	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 19.
ppi25_in_1	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 1.
ppi25_in_20	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 20.
ppi25_in_21	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 21.
ppi25_in_22	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 22.
ppi25_in_23	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 23.
ppi25_in_24	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 24.
ppi25_in_25	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 25.
ppi25_in_26	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 26.
ppi25_in_27	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 27.
ppi25_in_28	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 28.
ppi25_in_29	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 29.

Port	Direction	Protocol	Description
ppi25_in_2	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 2.
ppi25_in_30	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 30.
ppi25_in_31	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 31.
ppi25_in_3	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 3.
ppi25_in_4	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 4.
ppi25_in_5	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 5.
ppi25_in_6	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 6.
ppi25_in_7	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 7.
ppi25_in_8	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 8.
ppi25_in_9	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 9.
ppi26_in_0	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 0.
ppi26_in_10	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 10.
ppi26_in_11	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 11.
ppi26_in_12	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 12.
ppi26_in_13	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 13.
ppi26_in_14	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 14.
ppi26_in_15	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 15.
ppi26_in_16	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 16.
ppi26_in_17	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 17.
ppi26_in_18	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 18.
ppi26_in_19	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 19.
ppi26_in_1	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 1.
ppi26_in_20	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 20.
ppi26_in_21	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 21.
ppi26_in_22	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 22.
ppi26_in_23	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 23.
ppi26_in_24	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 24.
ppi26_in_25	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 25.
ppi26_in_26	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 26.
ppi26_in_27	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 27.
ppi26_in_28	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 28.
ppi26_in_29	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 29.
ppi26_in_2	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 2.

Port	Direction	Protocol	Description
ppi26_in_30	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 30.
ppi26_in_31	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 31.
ppi26_in_3	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 3.
ppi26_in_4	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 4.
ppi26_in_5	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 5.
ppi26_in_6	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 6.
ppi26_in_7	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 7.
ppi26_in_8	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 8.
ppi26_in_9	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 9.
ppi27_in_0	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 0.
ppi27_in_10	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 10.
ppi27_in_11	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 11.
ppi27_in_12	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 12.
ppi27_in_13	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 13.
ppi27_in_14	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 14.
ppi27_in_15	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 15.
ppi27_in_16	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 16.
ppi27_in_17	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 17.
ppi27_in_18	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 18.
ppi27_in_19	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 19.
ppi27_in_1	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 1.
ppi27_in_20	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 20.
ppi27_in_21	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 21.
ppi27_in_22	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 22.
ppi27_in_23	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 23.
ppi27_in_24	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 24.
ppi27_in_25	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 25.
ppi27_in_26	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 26.
ppi27_in_27	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 27.
ppi27_in_28	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 28.
ppi27_in_29	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 29.
ppi27_in_2	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 2.
ppi27_in_30	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 30.
ppi27_in_31	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 31.
ppi27_in_3	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 3.
ppi27_in_4	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 4.
ppi27_in_5	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 5.
ppi27_in_6	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 6.
ppi27_in_7	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 7.
ppi27_in_8	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 8.

Port	Direction	Protocol	Description
ppi27_in_9	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 9.
ppi28_in_0	slave	Signal	Private peripheral interrupts (ID28) of cluster 0.
ppi28_in_10	slave	Signal	Private peripheral interrupts (ID28) of cluster 10.
ppi28_in_11	slave	Signal	Private peripheral interrupts (ID28) of cluster 11.
ppi28_in_12	slave	Signal	Private peripheral interrupts (ID28) of cluster 12.
ppi28_in_13	slave	Signal	Private peripheral interrupts (ID28) of cluster 13.
ppi28_in_14	slave	Signal	Private peripheral interrupts (ID28) of cluster 14.
ppi28_in_15	slave	Signal	Private peripheral interrupts (ID28) of cluster 15.
ppi28_in_16	slave	Signal	Private peripheral interrupts (ID28) of cluster 16.
ppi28_in_17	slave	Signal	Private peripheral interrupts (ID28) of cluster 17.
ppi28_in_18	slave	Signal	Private peripheral interrupts (ID28) of cluster 18.
ppi28_in_19	slave	Signal	Private peripheral interrupts (ID28) of cluster 19.
ppi28_in_1	slave	Signal	Private peripheral interrupts (ID28) of cluster 1.
ppi28_in_20	slave	Signal	Private peripheral interrupts (ID28) of cluster 20.
ppi28_in_21	slave	Signal	Private peripheral interrupts (ID28) of cluster 21.
ppi28_in_22	slave	Signal	Private peripheral interrupts (ID28) of cluster 22.
ppi28_in_23	slave	Signal	Private peripheral interrupts (ID28) of cluster 23.
ppi28_in_24	slave	Signal	Private peripheral interrupts (ID28) of cluster 24.
ppi28_in_25	slave	Signal	Private peripheral interrupts (ID28) of cluster 25.
ppi28_in_26	slave	Signal	Private peripheral interrupts (ID28) of cluster 26.
ppi28_in_27	slave	Signal	Private peripheral interrupts (ID28) of cluster 27.
ppi28_in_28	slave	Signal	Private peripheral interrupts (ID28) of cluster 28.
ppi28_in_29	slave	Signal	Private peripheral interrupts (ID28) of cluster 29.
ppi28_in_2	slave	Signal	Private peripheral interrupts (ID28) of cluster 2.
ppi28_in_30	slave	Signal	Private peripheral interrupts (ID28) of cluster 30.
ppi28_in_31	slave	Signal	Private peripheral interrupts (ID28) of cluster 31.
ppi28_in_3	slave	Signal	Private peripheral interrupts (ID28) of cluster 3.
ppi28_in_4	slave	Signal	Private peripheral interrupts (ID28) of cluster 4.
ppi28_in_5	slave	Signal	Private peripheral interrupts (ID28) of cluster 5.
ppi28_in_6	slave	Signal	Private peripheral interrupts (ID28) of cluster 6.
ppi28_in_7	slave	Signal	Private peripheral interrupts (ID28) of cluster 7.
ppi28_in_8	slave	Signal	Private peripheral interrupts (ID28) of cluster 8.
ppi28_in_9	slave	Signal	Private peripheral interrupts (ID28) of cluster 9.
ppi29_in_0	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 0.
ppi29_in_10	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 10.
ppi29_in_11	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 11.
ppi29_in_12	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 12.
ppi29_in_13	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 13.
ppi29_in_14	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 14.
ppi29_in_15	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 15.

Port	Direction	Protocol	Description
ppi29_in_16	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 16.
ppi29_in_17	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 17.
ppi29_in_18	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 18.
ppi29_in_19	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 19.
ppi29_in_1	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 1.
ppi29_in_20	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 20.
ppi29_in_21	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 21.
ppi29_in_22	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 22.
ppi29_in_23	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 23.
ppi29_in_24	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 24.
ppi29_in_25	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 25.
ppi29_in_26	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 26.
ppi29_in_27	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 27.
ppi29_in_28	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 28.
ppi29_in_29	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 29.
ppi29_in_2	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 2.
ppi29_in_30	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 30.
ppi29_in_31	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 31.
ppi29_in_3	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 3.
ppi29_in_4	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 4.
ppi29_in_5	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 5.
ppi29_in_6	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 6.
ppi29_in_7	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 7.
ppi29_in_8	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 8.
ppi29_in_9	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 9.
ppi30_in_0	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 0.
ppi30_in_10	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 10.
ppi30_in_11	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 11.
ppi30_in_12	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 12.
ppi30_in_13	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 13.
ppi30_in_14	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 14.
ppi30_in_15	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 15.
ppi30_in_16	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 16.
ppi30_in_17	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 17.
ppi30_in_18	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 18.
ppi30_in_19	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 19.
ppi30_in_1	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 1.
ppi30_in_20	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 20.
ppi30_in_21	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 21.
ppi30_in_22	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 22.

Port	Direction	Protocol	Description
ppi30_in_23	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 23.
ppi30_in_24	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 24.
ppi30_in_25	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 25.
ppi30_in_26	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 26.
ppi30_in_27	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 27.
ppi30_in_28	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 28.
ppi30_in_29	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 29.
ppi30_in_2	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 2.
ppi30_in_30	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 30.
ppi30_in_31	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 31.
ppi30_in_3	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 3.
ppi30_in_4	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 4.
ppi30_in_5	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 5.
ppi30_in_6	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 6.
ppi30_in_7	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 7.
ppi30_in_8	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 8.
ppi30_in_9	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 9.
ppi31_in_0	slave	Signal	Private peripheral interrupts (ID31) of cluster 0.
ppi31_in_10	slave	Signal	Private peripheral interrupts (ID31) of cluster 10.
ppi31_in_11	slave	Signal	Private peripheral interrupts (ID31) of cluster 11.
ppi31_in_12	slave	Signal	Private peripheral interrupts (ID31) of cluster 12.
ppi31_in_13	slave	Signal	Private peripheral interrupts (ID31) of cluster 13.
ppi31_in_14	slave	Signal	Private peripheral interrupts (ID31) of cluster 14.
ppi31_in_15	slave	Signal	Private peripheral interrupts (ID31) of cluster 15.
ppi31_in_16	slave	Signal	Private peripheral interrupts (ID31) of cluster 16.
ppi31_in_17	slave	Signal	Private peripheral interrupts (ID31) of cluster 17.
ppi31_in_18	slave	Signal	Private peripheral interrupts (ID31) of cluster 18.
ppi31_in_19	slave	Signal	Private peripheral interrupts (ID31) of cluster 19.
ppi31_in_1	slave	Signal	Private peripheral interrupts (ID31) of cluster 1.
ppi31_in_20	slave	Signal	Private peripheral interrupts (ID31) of cluster 20.
ppi31_in_21	slave	Signal	Private peripheral interrupts (ID31) of cluster 21.
ppi31_in_22	slave	Signal	Private peripheral interrupts (ID31) of cluster 22.
ppi31_in_23	slave	Signal	Private peripheral interrupts (ID31) of cluster 23.
ppi31_in_24	slave	Signal	Private peripheral interrupts (ID31) of cluster 24.
ppi31_in_25	slave	Signal	Private peripheral interrupts (ID31) of cluster 25.
ppi31_in_26	slave	Signal	Private peripheral interrupts (ID31) of cluster 26.
ppi31_in_27	slave	Signal	Private peripheral interrupts (ID31) of cluster 27.
ppi31_in_28	slave	Signal	Private peripheral interrupts (ID31) of cluster 28.
ppi31_in_29	slave	Signal	Private peripheral interrupts (ID31) of cluster 29.
ppi31_in_2	slave	Signal	Private peripheral interrupts (ID31) of cluster 2.

Port	Direction	Protocol	Description
ppi31_in_30	slave	Signal	Private peripheral interrupts (ID31) of cluster 30.
ppi31_in_31	slave	Signal	Private peripheral interrupts (ID31) of cluster 31.
ppi31_in_3	slave	Signal	Private peripheral interrupts (ID31) of cluster 3.
ppi31_in_4	slave	Signal	Private peripheral interrupts (ID31) of cluster 4.
ppi31_in_5	slave	Signal	Private peripheral interrupts (ID31) of cluster 5.
ppi31_in_6	slave	Signal	Private peripheral interrupts (ID31) of cluster 6.
ppi31_in_7	slave	Signal	Private peripheral interrupts (ID31) of cluster 7.
ppi31_in_8	slave	Signal	Private peripheral interrupts (ID31) of cluster 8.
ppi31_in_9	slave	Signal	Private peripheral interrupts (ID31) of cluster 9.
pvbus_m	master	PVBus	Memory bus out: transactions generated by the IRI.
pvbus_s	slave	PVBus	Memory bus in: memory-mapped register accesses are accepted on this interface.
redistributor_m	master	GlCv3Comms	Input from and output to CPU interface.
reset	slave	Signal	Reset.
spi_in	slave	Signal	Shared peripheral interrupts.
wake_request_0	master	Signal	Power management outputs of cluster 0.
wake_request_10	master	Signal	Power management outputs of cluster 10.
wake_request_11	master	Signal	Power management outputs of cluster 11.
wake_request_12	master	Signal	Power management outputs of cluster 12.
wake_request_13	master	Signal	Power management outputs of cluster 13.
wake_request_14	master	Signal	Power management outputs of cluster 14.
wake_request_15	master	Signal	Power management outputs of cluster 15.
wake_request_16	master	Signal	Power management outputs of cluster 16.
wake_request_17	master	Signal	Power management outputs of cluster 17.
wake_request_18	master	Signal	Power management outputs of cluster 18.
wake_request_19	master	Signal	Power management outputs of cluster 19.
wake_request_1	master	Signal	Power management outputs of cluster 1.
wake_request_20	master	Signal	Power management outputs of cluster 20.
wake_request_21	master	Signal	Power management outputs of cluster 21.
wake_request_22	master	Signal	Power management outputs of cluster 22.
wake_request_23	master	Signal	Power management outputs of cluster 23.
wake_request_24	master	Signal	Power management outputs of cluster 24.
wake_request_25	master	Signal	Power management outputs of cluster 25.
wake_request_26	master	Signal	Power management outputs of cluster 26.
wake_request_27	master	Signal	Power management outputs of cluster 27.
wake_request_28	master	Signal	Power management outputs of cluster 28.
wake_request_29	master	Signal	Power management outputs of cluster 29.
wake_request_2	master	Signal	Power management outputs of cluster 2.
wake_request_30	master	Signal	Power management outputs of cluster 30.
wake_request_31	master	Signal	Power management outputs of cluster 31.
wake_request_3	master	Signal	Power management outputs of cluster 3.



Port	Direction	Protocol	Description
wake_request_4	master	Signal	Power management outputs of cluster 4.
wake_request_5	master	Signal	Power management outputs of cluster 5.
wake_request_6	master	Signal	Power management outputs of cluster 6.
wake_request_7	master	Signal	Power management outputs of cluster 7.
wake_request_8	master	Signal	Power management outputs of cluster 8.
wake_request_9	master	Signal	Power management outputs of cluster 9.

## Parameters for GIC500

### **GICD\_ITARGETSR-RAZWI**

If true, the GICD\_ITARGETS registers are **RAZ/WI**.

Type: `bool`

Default value: `false`

### **ITS-count**

Number of Interrupt Translation Services to be instantiated (0=none).

Type: `uint8_t`

Default value: `1`

### **ITS-device-bits**

Number of bits supported for ITS device IDs.

Type: `int`

Default value: `16`

### **ITS-threaded-command-queue**

Enable execution of ITS commands in a separate thread which is sometimes required for cosimulation.

Type: `bool`

Default value: `true`

### **SPI-count**

Number of SPIs that are implemented.

Type: `uint16_t`

Default value: `224`



**clear-ISPENDR-bit-for-level-sensitive-interrupt-when-acknowledged**

If true, acknowledgement of level sensitive interrupt clears the ISPENDR register.

Type: `bool`

Default value: `false`

**cpus\_per\_cluster\_0**

Number of cores within cluster 0.

Type: `uint8_t`

Default value: 1

**cpus\_per\_cluster\_1**

Number of cores within cluster 1.

Type: `uint8_t`

Default value: 1

**cpus\_per\_cluster\_10**

Number of cores within cluster 10.

Type: `uint8_t`

Default value: 1

**cpus\_per\_cluster\_11**

Number of cores within cluster 11.

Type: `uint8_t`

Default value: 1

**cpus\_per\_cluster\_12**

Number of cores within cluster 12.

Type: `uint8_t`

Default value: 1

**cpus\_per\_cluster\_13**

Number of cores within cluster 13.

Type: `uint8_t`

Default value: 1

**cpus\_per\_cluster\_14**

Number of cores within cluster 14.

Type: `uint8_t`

Default value: 1

**cpus\_per\_cluster\_15**

Number of cores within cluster 15.

Type: `uint8_t`

Default value: 1

**cpus\_per\_cluster\_16**

Number of cores within cluster 16.

Type: `uint8_t`

Default value: 1

**cpus\_per\_cluster\_17**

Number of cores within cluster 17.

Type: `uint8_t`

Default value: 1

**cpus\_per\_cluster\_18**

Number of cores within cluster 18.

Type: `uint8_t`

Default value: 1

**cpus\_per\_cluster\_19**

Number of cores within cluster 19.

Type: `uint8_t`

Default value: 1

**cpus\_per\_cluster\_2**

Number of cores within cluster 2.

Type: `uint8_t`

Default value: 1

**cpus\_per\_cluster\_20**

Number of cores within cluster 20.

Type: `uint8_t`

Default value: 1

**cpus\_per\_cluster\_21**

Number of cores within cluster 21.

Type: `uint8_t`

Default value: 1

**cpus\_per\_cluster\_22**

Number of cores within cluster 22.

Type: `uint8_t`

Default value: 1

**cpus\_per\_cluster\_23**

Number of cores within cluster 23.

Type: `uint8_t`

Default value: 1

**cpus\_per\_cluster\_24**

Number of cores within cluster 24.

Type: `uint8_t`

Default value: 1

**cpus\_per\_cluster\_25**

Number of cores within cluster 25.

Type: `uint8_t`

Default value: 1

**cpus\_per\_cluster\_26**

Number of cores within cluster 26.

Type: `uint8_t`

Default value: 1

**cpus\_per\_cluster\_27**

Number of cores within cluster 27.

Type: `uint8_t`

Default value: 1

**cpus\_per\_cluster\_28**

Number of cores within cluster 28.

Type: `uint8_t`

Default value: 1

**cpus\_per\_cluster\_29**

Number of cores within cluster 29.

Type: `uint8_t`

Default value: 1

**cpus\_per\_cluster\_3**

Number of cores within cluster 3.

Type: `uint8_t`

Default value: 1

**cpus\_per\_cluster\_30**

Number of cores within cluster 30.

Type: `uint8_t`

Default value: 1

**cpus\_per\_cluster\_31**

Number of cores within cluster 31.

Type: `uint8_t`

Default value: 1

**cpus\_per\_cluster\_4**

Number of cores within cluster 4.

Type: `uint8_t`

Default value: 1

**cpus\_per\_cluster\_5**

Number of cores within cluster 5.

Type: `uint8_t`

Default value: 1

**cpus\_per\_cluster\_6**

Number of cores within cluster 6.

Type: `uint8_t`

Default value: 1

**cpus\_per\_cluster\_7**

Number of cores within cluster 7.

Type: `uint8_t`

Default value: 1

**cpus\_per\_cluster\_8**

Number of cores within cluster 8.

Type: `uint8_t`

Default value: 1

**cpus\_per\_cluster\_9**

Number of cores within cluster 9.

Type: `uint8_t`

Default value: 1

**delay-ITS-accesses**

Delay accesses from the ITS until GICR\_SYNCR is read.

Type: `bool`

Default value: `true`

**delay-redistributor-accesses**

Delay memory accesses from the redistributor until GICR\_SYNCR is read.

Type: `bool`

Default value: `true`

**enable\_protocol\_checking**

Enable/disable protocol checking at cpu interface.

Type: `bool`

Default value: `false`

**enabled**

Enable GICv3 functionality; when false the component is inactive.

Type: `bool`

Default value: `true`

**has-two-security-states**

If true, has two security states.

Type: `bool`

Default value: `true`

**num\_clusters**

Number of implemented affinity level1 clusters.

Type: `uint8_t`

Default value: `1`

**print-memory-map**

Print memory map to stdout.

Type: `bool`

Default value: `false`

**redistributor-threaded-sync**

Enable execution of redistributor delayed transactions in a separate thread which is sometimes required for cosimulation.

Type: `bool`

Default value: `true`

**reg-base**

GIC500 base address.

Type: `uint64_t`

Default value: `0x2c010000`

### **using-generated-memorymap**

Use the generated memorymap for the GIC500 and warn if superfluous parameters are passed.

Type: `bool`

Default value: `true`

### **wakeup-on-reset**

Go against specification and start redistributors in woken-up state at reset. This allows software that was written for previous versions of the GICv3 specification to work correctly. This should not be used for production code or when the distributor is used separately from the core fast model.

Type: `bool`

Default value: `false`

## 3.186 GIC500\_ClusterPorts

Defined in `LISA/GIC500.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### **About GIC500\_ClusterPorts**

An alternate version of GIC500, identical to the above except for the CPUIF ports being exposed as an array per cluster

### **Iris and MTI instances for GIC500\_ClusterPorts**

This model has the following Iris instances:

Name	Instance type
<code>GIC500_ClusterPorts</code>	<code>GIC500</code>
<code>GIC500_ClusterPorts.ITS0</code>	<code>GICv3InterruptTranslationService</code>
<code>GIC500_ClusterPorts.ITS0.ExportTest.GIC500_ClusterPorts.ITS0.pvbus_m[0].pvbusmaster</code>	<code>PVBusMaster</code>
<code>GIC500_ClusterPorts.rd_0</code>	<code>GICv3RedistributorInternal</code>
<code>GIC500_ClusterPorts.rd_0_0</code>	<code>GICv3RedistributorInternal</code>
<code>GIC500_ClusterPorts.rd_0_0_0</code>	<code>GICv3RedistributorInternal</code>
<code>GIC500_ClusterPorts.rd_0_0_0_0</code>	<code>GICv3Redistributor</code>

Name	Instance type
GIC500_ClusterPorts.rd_0_0_0_0.ExportTest.GIC500_ClusterPorts.rd_0_0_0_0.pvbus_m[0].pvbusmaster	PVBusMaster
GIC500_ClusterPorts.rd_tl	GICv3Distributor

This model has the following MTI trace components:

Name	Component type
GIC500_ClusterPorts.ITS0	GICv3InterruptTranslationService
GIC500_ClusterPorts.ITS0.ExportTest.GIC500_ClusterPorts.ITS0.pvbus_m[0].pvbusmaster	PVBusMaster
GIC500_ClusterPorts.rd_0	GICv3RedistributorInternal
GIC500_ClusterPorts.rd_0_0	GICv3RedistributorInternal
GIC500_ClusterPorts.rd_0_0_0	GICv3RedistributorInternal
GIC500_ClusterPorts.rd_0_0_0_0	GICv3Redistributor
GIC500_ClusterPorts.rd_0_0_0_0.ExportTest.GIC500_ClusterPorts.rd_0_0_0_0.pvbus_m[0].pvbusmaster	PVBusMaster
GIC500_ClusterPorts.rd_tl	GICv3Distributor

### Ports for GIC500\_ClusterPorts

Port	Direction	Protocol	Description
cfgsdisable	slave	Signal	Disable some SPIs signal.
cpu_active_0	slave	Signal	cpu_active pins of cluster 0.
cpu_active_10	slave	Signal	cpu_active pins of cluster 10.
cpu_active_11	slave	Signal	cpu_active pins of cluster 11.
cpu_active_12	slave	Signal	cpu_active pins of cluster 12.
cpu_active_13	slave	Signal	cpu_active pins of cluster 13.
cpu_active_14	slave	Signal	cpu_active pins of cluster 14.
cpu_active_15	slave	Signal	cpu_active pins of cluster 15.
cpu_active_16	slave	Signal	cpu_active pins of cluster 16.
cpu_active_17	slave	Signal	cpu_active pins of cluster 17.
cpu_active_18	slave	Signal	cpu_active pins of cluster 18.
cpu_active_19	slave	Signal	cpu_active pins of cluster 19.
cpu_active_1	slave	Signal	cpu_active pins of cluster 1.
cpu_active_20	slave	Signal	cpu_active pins of cluster 20.
cpu_active_21	slave	Signal	cpu_active pins of cluster 21.
cpu_active_22	slave	Signal	cpu_active pins of cluster 22.
cpu_active_23	slave	Signal	cpu_active pins of cluster 23.
cpu_active_24	slave	Signal	cpu_active pins of cluster 24.
cpu_active_25	slave	Signal	cpu_active pins of cluster 25.
cpu_active_26	slave	Signal	cpu_active pins of cluster 26.
cpu_active_27	slave	Signal	cpu_active pins of cluster 27.
cpu_active_28	slave	Signal	cpu_active pins of cluster 28.



Port	Direction	Protocol	Description
cpu_active_29	slave	Signal	cpu_active pins of cluster 29.
cpu_active_2	slave	Signal	cpu_active pins of cluster 2.
cpu_active_30	slave	Signal	cpu_active pins of cluster 30.
cpu_active_31	slave	Signal	cpu_active pins of cluster 31.
cpu_active_3	slave	Signal	cpu_active pins of cluster 3.
cpu_active_4	slave	Signal	cpu_active pins of cluster 4.
cpu_active_5	slave	Signal	cpu_active pins of cluster 5.
cpu_active_6	slave	Signal	cpu_active pins of cluster 6.
cpu_active_7	slave	Signal	cpu_active pins of cluster 7.
cpu_active_8	slave	Signal	cpu_active pins of cluster 8.
cpu_active_9	slave	Signal	cpu_active pins of cluster 9.
po_reset	slave	Signal	Power on reset.
ppi16_in_0	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 0.
ppi16_in_10	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 10.
ppi16_in_11	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 11.
ppi16_in_12	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 12.
ppi16_in_13	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 13.
ppi16_in_14	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 14.
ppi16_in_15	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 15.
ppi16_in_16	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 16.
ppi16_in_17	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 17.
ppi16_in_18	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 18.
ppi16_in_19	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 19.
ppi16_in_1	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 1.
ppi16_in_20	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 20.
ppi16_in_21	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 21.
ppi16_in_22	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 22.
ppi16_in_23	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 23.
ppi16_in_24	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 24.
ppi16_in_25	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 25.
ppi16_in_26	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 26.
ppi16_in_27	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 27.
ppi16_in_28	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 28.
ppi16_in_29	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 29.
ppi16_in_2	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 2.
ppi16_in_30	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 30.
ppi16_in_31	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 31.
ppi16_in_3	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 3.
ppi16_in_4	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 4.
ppi16_in_5	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 5.

Port	Direction	Protocol	Description
ppi16_in_6	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 6.
ppi16_in_7	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 7.
ppi16_in_8	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 8.
ppi16_in_9	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 9.
ppi17_in_0	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 0.
ppi17_in_10	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 10.
ppi17_in_11	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 11.
ppi17_in_12	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 12.
ppi17_in_13	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 13.
ppi17_in_14	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 14.
ppi17_in_15	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 15.
ppi17_in_16	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 16.
ppi17_in_17	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 17.
ppi17_in_18	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 18.
ppi17_in_19	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 19.
ppi17_in_1	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 1.
ppi17_in_20	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 20.
ppi17_in_21	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 21.
ppi17_in_22	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 22.
ppi17_in_23	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 23.
ppi17_in_24	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 24.
ppi17_in_25	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 25.
ppi17_in_26	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 26.
ppi17_in_27	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 27.
ppi17_in_28	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 28.
ppi17_in_29	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 29.
ppi17_in_2	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 2.
ppi17_in_30	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 30.
ppi17_in_31	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 31.
ppi17_in_3	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 3.
ppi17_in_4	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 4.
ppi17_in_5	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 5.
ppi17_in_6	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 6.
ppi17_in_7	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 7.
ppi17_in_8	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 8.
ppi17_in_9	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 9.
ppi18_in_0	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 0.
ppi18_in_10	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 10.
ppi18_in_11	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 11.
ppi18_in_12	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 12.

Port	Direction	Protocol	Description
ppi18_in_13	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 13.
ppi18_in_14	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 14.
ppi18_in_15	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 15.
ppi18_in_16	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 16.
ppi18_in_17	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 17.
ppi18_in_18	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 18.
ppi18_in_19	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 19.
ppi18_in_1	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 1.
ppi18_in_20	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 20.
ppi18_in_21	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 21.
ppi18_in_22	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 22.
ppi18_in_23	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 23.
ppi18_in_24	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 24.
ppi18_in_25	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 25.
ppi18_in_26	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 26.
ppi18_in_27	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 27.
ppi18_in_28	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 28.
ppi18_in_29	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 29.
ppi18_in_2	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 2.
ppi18_in_30	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 30.
ppi18_in_31	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 31.
ppi18_in_3	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 3.
ppi18_in_4	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 4.
ppi18_in_5	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 5.
ppi18_in_6	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 6.
ppi18_in_7	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 7.
ppi18_in_8	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 8.
ppi18_in_9	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 9.
ppi19_in_0	slave	Signal	Private peripheral interrupts (ID19) of cluster 0.
ppi19_in_10	slave	Signal	Private peripheral interrupts (ID19) of cluster 10.
ppi19_in_11	slave	Signal	Private peripheral interrupts (ID19) of cluster 11.
ppi19_in_12	slave	Signal	Private peripheral interrupts (ID19) of cluster 12.
ppi19_in_13	slave	Signal	Private peripheral interrupts (ID19) of cluster 13.
ppi19_in_14	slave	Signal	Private peripheral interrupts (ID19) of cluster 14.
ppi19_in_15	slave	Signal	Private peripheral interrupts (ID19) of cluster 15.
ppi19_in_16	slave	Signal	Private peripheral interrupts (ID19) of cluster 16.
ppi19_in_17	slave	Signal	Private peripheral interrupts (ID19) of cluster 17.
ppi19_in_18	slave	Signal	Private peripheral interrupts (ID19) of cluster 18.
ppi19_in_19	slave	Signal	Private peripheral interrupts (ID19) of cluster 19.
ppi19_in_1	slave	Signal	Private peripheral interrupts (ID19) of cluster 1.

Port	Direction	Protocol	Description
ppi19_in_20	slave	Signal	Private peripheral interrupts (ID19) of cluster 20.
ppi19_in_21	slave	Signal	Private peripheral interrupts (ID19) of cluster 21.
ppi19_in_22	slave	Signal	Private peripheral interrupts (ID19) of cluster 22.
ppi19_in_23	slave	Signal	Private peripheral interrupts (ID19) of cluster 23.
ppi19_in_24	slave	Signal	Private peripheral interrupts (ID19) of cluster 24.
ppi19_in_25	slave	Signal	Private peripheral interrupts (ID19) of cluster 25.
ppi19_in_26	slave	Signal	Private peripheral interrupts (ID19) of cluster 26.
ppi19_in_27	slave	Signal	Private peripheral interrupts (ID19) of cluster 27.
ppi19_in_28	slave	Signal	Private peripheral interrupts (ID19) of cluster 28.
ppi19_in_29	slave	Signal	Private peripheral interrupts (ID19) of cluster 29.
ppi19_in_2	slave	Signal	Private peripheral interrupts (ID19) of cluster 2.
ppi19_in_30	slave	Signal	Private peripheral interrupts (ID19) of cluster 30.
ppi19_in_31	slave	Signal	Private peripheral interrupts (ID19) of cluster 31.
ppi19_in_3	slave	Signal	Private peripheral interrupts (ID19) of cluster 3.
ppi19_in_4	slave	Signal	Private peripheral interrupts (ID19) of cluster 4.
ppi19_in_5	slave	Signal	Private peripheral interrupts (ID19) of cluster 5.
ppi19_in_6	slave	Signal	Private peripheral interrupts (ID19) of cluster 6.
ppi19_in_7	slave	Signal	Private peripheral interrupts (ID19) of cluster 7.
ppi19_in_8	slave	Signal	Private peripheral interrupts (ID19) of cluster 8.
ppi19_in_9	slave	Signal	Private peripheral interrupts (ID19) of cluster 9.
ppi20_in_0	slave	Signal	Private peripheral interrupts (ID20) of cluster 0.
ppi20_in_10	slave	Signal	Private peripheral interrupts (ID20) of cluster 10.
ppi20_in_11	slave	Signal	Private peripheral interrupts (ID20) of cluster 11.
ppi20_in_12	slave	Signal	Private peripheral interrupts (ID20) of cluster 12.
ppi20_in_13	slave	Signal	Private peripheral interrupts (ID20) of cluster 13.
ppi20_in_14	slave	Signal	Private peripheral interrupts (ID20) of cluster 14.
ppi20_in_15	slave	Signal	Private peripheral interrupts (ID20) of cluster 15.
ppi20_in_16	slave	Signal	Private peripheral interrupts (ID20) of cluster 16.
ppi20_in_17	slave	Signal	Private peripheral interrupts (ID20) of cluster 17.
ppi20_in_18	slave	Signal	Private peripheral interrupts (ID20) of cluster 18.
ppi20_in_19	slave	Signal	Private peripheral interrupts (ID20) of cluster 19.
ppi20_in_1	slave	Signal	Private peripheral interrupts (ID20) of cluster 1.
ppi20_in_20	slave	Signal	Private peripheral interrupts (ID20) of cluster 20.
ppi20_in_21	slave	Signal	Private peripheral interrupts (ID20) of cluster 21.
ppi20_in_22	slave	Signal	Private peripheral interrupts (ID20) of cluster 22.
ppi20_in_23	slave	Signal	Private peripheral interrupts (ID20) of cluster 23.
ppi20_in_24	slave	Signal	Private peripheral interrupts (ID20) of cluster 24.
ppi20_in_25	slave	Signal	Private peripheral interrupts (ID20) of cluster 25.
ppi20_in_26	slave	Signal	Private peripheral interrupts (ID20) of cluster 26.
ppi20_in_27	slave	Signal	Private peripheral interrupts (ID20) of cluster 27.

Port	Direction	Protocol	Description
ppi20_in_28	slave	Signal	Private peripheral interrupts (ID20) of cluster 28.
ppi20_in_29	slave	Signal	Private peripheral interrupts (ID20) of cluster 29.
ppi20_in_2	slave	Signal	Private peripheral interrupts (ID20) of cluster 2.
ppi20_in_30	slave	Signal	Private peripheral interrupts (ID20) of cluster 30.
ppi20_in_31	slave	Signal	Private peripheral interrupts (ID20) of cluster 31.
ppi20_in_3	slave	Signal	Private peripheral interrupts (ID20) of cluster 3.
ppi20_in_4	slave	Signal	Private peripheral interrupts (ID20) of cluster 4.
ppi20_in_5	slave	Signal	Private peripheral interrupts (ID20) of cluster 5.
ppi20_in_6	slave	Signal	Private peripheral interrupts (ID20) of cluster 6.
ppi20_in_7	slave	Signal	Private peripheral interrupts (ID20) of cluster 7.
ppi20_in_8	slave	Signal	Private peripheral interrupts (ID20) of cluster 8.
ppi20_in_9	slave	Signal	Private peripheral interrupts (ID20) of cluster 9.
ppi21_in_0	slave	Signal	Private peripheral interrupts (ID21) of cluster 0.
ppi21_in_10	slave	Signal	Private peripheral interrupts (ID21) of cluster 10.
ppi21_in_11	slave	Signal	Private peripheral interrupts (ID21) of cluster 11.
ppi21_in_12	slave	Signal	Private peripheral interrupts (ID21) of cluster 12.
ppi21_in_13	slave	Signal	Private peripheral interrupts (ID21) of cluster 13.
ppi21_in_14	slave	Signal	Private peripheral interrupts (ID21) of cluster 14.
ppi21_in_15	slave	Signal	Private peripheral interrupts (ID21) of cluster 15.
ppi21_in_16	slave	Signal	Private peripheral interrupts (ID21) of cluster 16.
ppi21_in_17	slave	Signal	Private peripheral interrupts (ID21) of cluster 17.
ppi21_in_18	slave	Signal	Private peripheral interrupts (ID21) of cluster 18.
ppi21_in_19	slave	Signal	Private peripheral interrupts (ID21) of cluster 19.
ppi21_in_1	slave	Signal	Private peripheral interrupts (ID21) of cluster 1.
ppi21_in_20	slave	Signal	Private peripheral interrupts (ID21) of cluster 20.
ppi21_in_21	slave	Signal	Private peripheral interrupts (ID21) of cluster 21.
ppi21_in_22	slave	Signal	Private peripheral interrupts (ID21) of cluster 22.
ppi21_in_23	slave	Signal	Private peripheral interrupts (ID21) of cluster 23.
ppi21_in_24	slave	Signal	Private peripheral interrupts (ID21) of cluster 24.
ppi21_in_25	slave	Signal	Private peripheral interrupts (ID21) of cluster 25.
ppi21_in_26	slave	Signal	Private peripheral interrupts (ID21) of cluster 26.
ppi21_in_27	slave	Signal	Private peripheral interrupts (ID21) of cluster 27.
ppi21_in_28	slave	Signal	Private peripheral interrupts (ID21) of cluster 28.
ppi21_in_29	slave	Signal	Private peripheral interrupts (ID21) of cluster 29.
ppi21_in_2	slave	Signal	Private peripheral interrupts (ID21) of cluster 2.
ppi21_in_30	slave	Signal	Private peripheral interrupts (ID21) of cluster 30.
ppi21_in_31	slave	Signal	Private peripheral interrupts (ID21) of cluster 31.
ppi21_in_3	slave	Signal	Private peripheral interrupts (ID21) of cluster 3.
ppi21_in_4	slave	Signal	Private peripheral interrupts (ID21) of cluster 4.
ppi21_in_5	slave	Signal	Private peripheral interrupts (ID21) of cluster 5.

Port	Direction	Protocol	Description
ppi21_in_6	slave	Signal	Private peripheral interrupts (ID21) of cluster 6.
ppi21_in_7	slave	Signal	Private peripheral interrupts (ID21) of cluster 7.
ppi21_in_8	slave	Signal	Private peripheral interrupts (ID21) of cluster 8.
ppi21_in_9	slave	Signal	Private peripheral interrupts (ID21) of cluster 9.
ppi22_in_0	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 0.
ppi22_in_10	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 10.
ppi22_in_11	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 11.
ppi22_in_12	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 12.
ppi22_in_13	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 13.
ppi22_in_14	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 14.
ppi22_in_15	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 15.
ppi22_in_16	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 16.
ppi22_in_17	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 17.
ppi22_in_18	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 18.
ppi22_in_19	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 19.
ppi22_in_1	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 1.
ppi22_in_20	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 20.
ppi22_in_21	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 21.
ppi22_in_22	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 22.
ppi22_in_23	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 23.
ppi22_in_24	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 24.
ppi22_in_25	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 25.
ppi22_in_26	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 26.
ppi22_in_27	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 27.
ppi22_in_28	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 28.

Port	Direction	Protocol	Description
ppi22_in_29	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 29.
ppi22_in_2	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 2.
ppi22_in_30	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 30.
ppi22_in_31	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 31.
ppi22_in_3	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 3.
ppi22_in_4	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 4.
ppi22_in_5	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 5.
ppi22_in_6	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 6.
ppi22_in_7	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 7.
ppi22_in_8	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 8.
ppi22_in_9	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 9.
ppi23_in_0	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 0.
ppi23_in_10	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 10.
ppi23_in_11	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 11.
ppi23_in_12	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 12.
ppi23_in_13	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 13.
ppi23_in_14	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 14.
ppi23_in_15	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 15.
ppi23_in_16	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 16.
ppi23_in_17	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 17.
ppi23_in_18	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 18.
ppi23_in_19	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 19.
ppi23_in_1	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 1.



Port	Direction	Protocol	Description
ppi23_in_20	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 20.
ppi23_in_21	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 21.
ppi23_in_22	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 22.
ppi23_in_23	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 23.
ppi23_in_24	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 24.
ppi23_in_25	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 25.
ppi23_in_26	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 26.
ppi23_in_27	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 27.
ppi23_in_28	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 28.
ppi23_in_29	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 29.
ppi23_in_2	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 2.
ppi23_in_30	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 30.
ppi23_in_31	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 31.
ppi23_in_3	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 3.
ppi23_in_4	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 4.
ppi23_in_5	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 5.
ppi23_in_6	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 6.
ppi23_in_7	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 7.
ppi23_in_8	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 8.
ppi23_in_9	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 9.
ppi24_in_0	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 0.
ppi24_in_10	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 10.
ppi24_in_11	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 11.



Port	Direction	Protocol	Description
ppi24_in_12	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 12.
ppi24_in_13	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 13.
ppi24_in_14	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 14.
ppi24_in_15	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 15.
ppi24_in_16	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 16.
ppi24_in_17	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 17.
ppi24_in_18	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 18.
ppi24_in_19	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 19.
ppi24_in_1	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 1.
ppi24_in_20	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 20.
ppi24_in_21	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 21.
ppi24_in_22	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 22.
ppi24_in_23	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 23.
ppi24_in_24	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 24.
ppi24_in_25	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 25.
ppi24_in_26	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 26.
ppi24_in_27	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 27.
ppi24_in_28	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 28.
ppi24_in_29	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 29.
ppi24_in_2	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 2.
ppi24_in_30	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 30.
ppi24_in_31	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 31.
ppi24_in_3	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 3.

Port	Direction	Protocol	Description
ppi24_in_4	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 4.
ppi24_in_5	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 5.
ppi24_in_6	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 6.
ppi24_in_7	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 7.
ppi24_in_8	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 8.
ppi24_in_9	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 9.
ppi25_in_0	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 0.
ppi25_in_10	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 10.
ppi25_in_11	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 11.
ppi25_in_12	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 12.
ppi25_in_13	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 13.
ppi25_in_14	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 14.
ppi25_in_15	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 15.
ppi25_in_16	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 16.
ppi25_in_17	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 17.
ppi25_in_18	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 18.
ppi25_in_19	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 19.
ppi25_in_1	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 1.
ppi25_in_20	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 20.
ppi25_in_21	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 21.
ppi25_in_22	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 22.
ppi25_in_23	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 23.
ppi25_in_24	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 24.

Port	Direction	Protocol	Description
ppi25_in_25	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 25.
ppi25_in_26	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 26.
ppi25_in_27	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 27.
ppi25_in_28	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 28.
ppi25_in_29	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 29.
ppi25_in_2	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 2.
ppi25_in_30	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 30.
ppi25_in_31	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 31.
ppi25_in_3	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 3.
ppi25_in_4	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 4.
ppi25_in_5	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 5.
ppi25_in_6	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 6.
ppi25_in_7	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 7.
ppi25_in_8	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 8.
ppi25_in_9	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 9.
ppi26_in_0	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 0.
ppi26_in_10	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 10.
ppi26_in_11	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 11.
ppi26_in_12	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 12.
ppi26_in_13	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 13.
ppi26_in_14	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 14.
ppi26_in_15	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 15.
ppi26_in_16	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 16.
ppi26_in_17	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 17.
ppi26_in_18	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 18.
ppi26_in_19	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 19.
ppi26_in_1	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 1.
ppi26_in_20	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 20.
ppi26_in_21	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 21.
ppi26_in_22	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 22.

Port	Direction	Protocol	Description
ppi26_in_23	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 23.
ppi26_in_24	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 24.
ppi26_in_25	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 25.
ppi26_in_26	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 26.
ppi26_in_27	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 27.
ppi26_in_28	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 28.
ppi26_in_29	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 29.
ppi26_in_2	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 2.
ppi26_in_30	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 30.
ppi26_in_31	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 31.
ppi26_in_3	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 3.
ppi26_in_4	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 4.
ppi26_in_5	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 5.
ppi26_in_6	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 6.
ppi26_in_7	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 7.
ppi26_in_8	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 8.
ppi26_in_9	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 9.
ppi27_in_0	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 0.
ppi27_in_10	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 10.
ppi27_in_11	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 11.
ppi27_in_12	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 12.
ppi27_in_13	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 13.
ppi27_in_14	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 14.
ppi27_in_15	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 15.
ppi27_in_16	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 16.
ppi27_in_17	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 17.
ppi27_in_18	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 18.
ppi27_in_19	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 19.
ppi27_in_1	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 1.
ppi27_in_20	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 20.
ppi27_in_21	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 21.
ppi27_in_22	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 22.
ppi27_in_23	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 23.
ppi27_in_24	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 24.
ppi27_in_25	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 25.
ppi27_in_26	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 26.
ppi27_in_27	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 27.
ppi27_in_28	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 28.
ppi27_in_29	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 29.
ppi27_in_2	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 2.

Port	Direction	Protocol	Description
ppi27_in_30	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 30.
ppi27_in_31	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 31.
ppi27_in_3	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 3.
ppi27_in_4	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 4.
ppi27_in_5	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 5.
ppi27_in_6	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 6.
ppi27_in_7	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 7.
ppi27_in_8	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 8.
ppi27_in_9	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 9.
ppi28_in_0	slave	Signal	Private peripheral interrupts (ID28) of cluster 0.
ppi28_in_10	slave	Signal	Private peripheral interrupts (ID28) of cluster 10.
ppi28_in_11	slave	Signal	Private peripheral interrupts (ID28) of cluster 11.
ppi28_in_12	slave	Signal	Private peripheral interrupts (ID28) of cluster 12.
ppi28_in_13	slave	Signal	Private peripheral interrupts (ID28) of cluster 13.
ppi28_in_14	slave	Signal	Private peripheral interrupts (ID28) of cluster 14.
ppi28_in_15	slave	Signal	Private peripheral interrupts (ID28) of cluster 15.
ppi28_in_16	slave	Signal	Private peripheral interrupts (ID28) of cluster 16.
ppi28_in_17	slave	Signal	Private peripheral interrupts (ID28) of cluster 17.
ppi28_in_18	slave	Signal	Private peripheral interrupts (ID28) of cluster 18.
ppi28_in_19	slave	Signal	Private peripheral interrupts (ID28) of cluster 19.
ppi28_in_1	slave	Signal	Private peripheral interrupts (ID28) of cluster 1.
ppi28_in_20	slave	Signal	Private peripheral interrupts (ID28) of cluster 20.
ppi28_in_21	slave	Signal	Private peripheral interrupts (ID28) of cluster 21.
ppi28_in_22	slave	Signal	Private peripheral interrupts (ID28) of cluster 22.
ppi28_in_23	slave	Signal	Private peripheral interrupts (ID28) of cluster 23.
ppi28_in_24	slave	Signal	Private peripheral interrupts (ID28) of cluster 24.
ppi28_in_25	slave	Signal	Private peripheral interrupts (ID28) of cluster 25.
ppi28_in_26	slave	Signal	Private peripheral interrupts (ID28) of cluster 26.
ppi28_in_27	slave	Signal	Private peripheral interrupts (ID28) of cluster 27.
ppi28_in_28	slave	Signal	Private peripheral interrupts (ID28) of cluster 28.
ppi28_in_29	slave	Signal	Private peripheral interrupts (ID28) of cluster 29.
ppi28_in_2	slave	Signal	Private peripheral interrupts (ID28) of cluster 2.
ppi28_in_30	slave	Signal	Private peripheral interrupts (ID28) of cluster 30.
ppi28_in_31	slave	Signal	Private peripheral interrupts (ID28) of cluster 31.
ppi28_in_3	slave	Signal	Private peripheral interrupts (ID28) of cluster 3.
ppi28_in_4	slave	Signal	Private peripheral interrupts (ID28) of cluster 4.
ppi28_in_5	slave	Signal	Private peripheral interrupts (ID28) of cluster 5.
ppi28_in_6	slave	Signal	Private peripheral interrupts (ID28) of cluster 6.
ppi28_in_7	slave	Signal	Private peripheral interrupts (ID28) of cluster 7.
ppi28_in_8	slave	Signal	Private peripheral interrupts (ID28) of cluster 8.

Port	Direction	Protocol	Description
ppi28_in_9	slave	Signal	Private peripheral interrupts (ID28) of cluster 9.
ppi29_in_0	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 0.
ppi29_in_10	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 10.
ppi29_in_11	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 11.
ppi29_in_12	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 12.
ppi29_in_13	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 13.
ppi29_in_14	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 14.
ppi29_in_15	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 15.
ppi29_in_16	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 16.
ppi29_in_17	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 17.
ppi29_in_18	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 18.
ppi29_in_19	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 19.
ppi29_in_1	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 1.
ppi29_in_20	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 20.
ppi29_in_21	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 21.
ppi29_in_22	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 22.
ppi29_in_23	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 23.
ppi29_in_24	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 24.
ppi29_in_25	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 25.
ppi29_in_26	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 26.
ppi29_in_27	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 27.
ppi29_in_28	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 28.
ppi29_in_29	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 29.
ppi29_in_2	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 2.
ppi29_in_30	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 30.
ppi29_in_31	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 31.
ppi29_in_3	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 3.
ppi29_in_4	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 4.
ppi29_in_5	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 5.
ppi29_in_6	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 6.
ppi29_in_7	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 7.
ppi29_in_8	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 8.
ppi29_in_9	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 9.
ppi30_in_0	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 0.
ppi30_in_10	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 10.
ppi30_in_11	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 11.
ppi30_in_12	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 12.

Port	Direction	Protocol	Description
ppi30_in_13	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 13.
ppi30_in_14	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 14.
ppi30_in_15	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 15.
ppi30_in_16	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 16.
ppi30_in_17	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 17.
ppi30_in_18	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 18.
ppi30_in_19	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 19.
ppi30_in_1	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 1.
ppi30_in_20	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 20.
ppi30_in_21	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 21.
ppi30_in_22	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 22.
ppi30_in_23	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 23.
ppi30_in_24	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 24.
ppi30_in_25	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 25.
ppi30_in_26	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 26.
ppi30_in_27	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 27.
ppi30_in_28	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 28.
ppi30_in_29	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 29.
ppi30_in_2	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 2.
ppi30_in_30	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 30.
ppi30_in_31	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 31.
ppi30_in_3	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 3.
ppi30_in_4	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 4.
ppi30_in_5	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 5.
ppi30_in_6	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 6.
ppi30_in_7	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 7.
ppi30_in_8	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 8.



Port	Direction	Protocol	Description
ppi30_in_9	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 9.
ppi31_in_0	slave	Signal	Private peripheral interrupts (ID31) of cluster 0.
ppi31_in_10	slave	Signal	Private peripheral interrupts (ID31) of cluster 10.
ppi31_in_11	slave	Signal	Private peripheral interrupts (ID31) of cluster 11.
ppi31_in_12	slave	Signal	Private peripheral interrupts (ID31) of cluster 12.
ppi31_in_13	slave	Signal	Private peripheral interrupts (ID31) of cluster 13.
ppi31_in_14	slave	Signal	Private peripheral interrupts (ID31) of cluster 14.
ppi31_in_15	slave	Signal	Private peripheral interrupts (ID31) of cluster 15.
ppi31_in_16	slave	Signal	Private peripheral interrupts (ID31) of cluster 16.
ppi31_in_17	slave	Signal	Private peripheral interrupts (ID31) of cluster 17.
ppi31_in_18	slave	Signal	Private peripheral interrupts (ID31) of cluster 18.
ppi31_in_19	slave	Signal	Private peripheral interrupts (ID31) of cluster 19.
ppi31_in_1	slave	Signal	Private peripheral interrupts (ID31) of cluster 1.
ppi31_in_20	slave	Signal	Private peripheral interrupts (ID31) of cluster 20.
ppi31_in_21	slave	Signal	Private peripheral interrupts (ID31) of cluster 21.
ppi31_in_22	slave	Signal	Private peripheral interrupts (ID31) of cluster 22.
ppi31_in_23	slave	Signal	Private peripheral interrupts (ID31) of cluster 23.
ppi31_in_24	slave	Signal	Private peripheral interrupts (ID31) of cluster 24.
ppi31_in_25	slave	Signal	Private peripheral interrupts (ID31) of cluster 25.
ppi31_in_26	slave	Signal	Private peripheral interrupts (ID31) of cluster 26.
ppi31_in_27	slave	Signal	Private peripheral interrupts (ID31) of cluster 27.
ppi31_in_28	slave	Signal	Private peripheral interrupts (ID31) of cluster 28.
ppi31_in_29	slave	Signal	Private peripheral interrupts (ID31) of cluster 29.
ppi31_in_2	slave	Signal	Private peripheral interrupts (ID31) of cluster 2.
ppi31_in_30	slave	Signal	Private peripheral interrupts (ID31) of cluster 30.
ppi31_in_31	slave	Signal	Private peripheral interrupts (ID31) of cluster 31.
ppi31_in_3	slave	Signal	Private peripheral interrupts (ID31) of cluster 3.
ppi31_in_4	slave	Signal	Private peripheral interrupts (ID31) of cluster 4.
ppi31_in_5	slave	Signal	Private peripheral interrupts (ID31) of cluster 5.
ppi31_in_6	slave	Signal	Private peripheral interrupts (ID31) of cluster 6.
ppi31_in_7	slave	Signal	Private peripheral interrupts (ID31) of cluster 7.
ppi31_in_8	slave	Signal	Private peripheral interrupts (ID31) of cluster 8.
ppi31_in_9	slave	Signal	Private peripheral interrupts (ID31) of cluster 9.
pvbuss_m	master	PVBus	Memory bus out: transactions generated by the IRI.
pvbuss_s	slave	PVBus	Memory bus in: memory-mapped register accesses are accepted on this interface.
redistributor_0	master	GlCv3Comms	Input from and output to CPU interface for cluster 0
redistributor_10	master	GlCv3Comms	Input from and output to CPU interface for cluster 10.
redistributor_11	master	GlCv3Comms	Input from and output to CPU interface for cluster 11.
redistributor_12	master	GlCv3Comms	Input from and output to CPU interface for cluster 12.
redistributor_13	master	GlCv3Comms	Input from and output to CPU interface for cluster 13.



Port	Direction	Protocol	Description
redistributor_14	master	GlCv3Comms	Input from and output to CPU interface for cluster 14.
redistributor_15	master	GlCv3Comms	Input from and output to CPU interface for cluster 15.
redistributor_16	master	GlCv3Comms	Input from and output to CPU interface for cluster 16.
redistributor_17	master	GlCv3Comms	Input from and output to CPU interface for cluster 17.
redistributor_18	master	GlCv3Comms	Input from and output to CPU interface for cluster 18.
redistributor_19	master	GlCv3Comms	Input from and output to CPU interface for cluster 19.
redistributor_1	master	GlCv3Comms	Input from and output to CPU interface for cluster 1.
redistributor_20	master	GlCv3Comms	Input from and output to CPU interface for cluster 20.
redistributor_21	master	GlCv3Comms	Input from and output to CPU interface for cluster 21.
redistributor_22	master	GlCv3Comms	Input from and output to CPU interface for cluster 22.
redistributor_23	master	GlCv3Comms	Input from and output to CPU interface for cluster 23.
redistributor_24	master	GlCv3Comms	Input from and output to CPU interface for cluster 24.
redistributor_25	master	GlCv3Comms	Input from and output to CPU interface for cluster 25.
redistributor_26	master	GlCv3Comms	Input from and output to CPU interface for cluster 26.
redistributor_27	master	GlCv3Comms	Input from and output to CPU interface for cluster 27.
redistributor_28	master	GlCv3Comms	Input from and output to CPU interface for cluster 28.
redistributor_29	master	GlCv3Comms	Input from and output to CPU interface for cluster 29.
redistributor_2	master	GlCv3Comms	Input from and output to CPU interface for cluster 2.
redistributor_30	master	GlCv3Comms	Input from and output to CPU interface for cluster 30.
redistributor_31	master	GlCv3Comms	Input from and output to CPU interface for cluster 31.
redistributor_3	master	GlCv3Comms	Input from and output to CPU interface for cluster 3.
redistributor_4	master	GlCv3Comms	Input from and output to CPU interface for cluster 4.
redistributor_5	master	GlCv3Comms	Input from and output to CPU interface for cluster 5.
redistributor_6	master	GlCv3Comms	Input from and output to CPU interface for cluster 6.
redistributor_7	master	GlCv3Comms	Input from and output to CPU interface for cluster 7.
redistributor_8	master	GlCv3Comms	Input from and output to CPU interface for cluster 8.
redistributor_9	master	GlCv3Comms	Input from and output to CPU interface for cluster 9.
reset	slave	Signal	Reset.
spi_in	slave	Signal	Shared peripheral interrupts.
wake_request_0	master	Signal	Power management outputs of cluster 0.
wake_request_10	master	Signal	Power management outputs of cluster 10.
wake_request_11	master	Signal	Power management outputs of cluster 11.
wake_request_12	master	Signal	Power management outputs of cluster 12.
wake_request_13	master	Signal	Power management outputs of cluster 13.
wake_request_14	master	Signal	Power management outputs of cluster 14.
wake_request_15	master	Signal	Power management outputs of cluster 15.
wake_request_16	master	Signal	Power management outputs of cluster 16.
wake_request_17	master	Signal	Power management outputs of cluster 17.
wake_request_18	master	Signal	Power management outputs of cluster 18.
wake_request_19	master	Signal	Power management outputs of cluster 19.

Port	Direction	Protocol	Description
wake_request_1	master	Signal	Power management outputs of cluster 1.
wake_request_20	master	Signal	Power management outputs of cluster 20.
wake_request_21	master	Signal	Power management outputs of cluster 21.
wake_request_22	master	Signal	Power management outputs of cluster 22.
wake_request_23	master	Signal	Power management outputs of cluster 23.
wake_request_24	master	Signal	Power management outputs of cluster 24.
wake_request_25	master	Signal	Power management outputs of cluster 25.
wake_request_26	master	Signal	Power management outputs of cluster 26.
wake_request_27	master	Signal	Power management outputs of cluster 27.
wake_request_28	master	Signal	Power management outputs of cluster 28.
wake_request_29	master	Signal	Power management outputs of cluster 29.
wake_request_2	master	Signal	Power management outputs of cluster 2.
wake_request_30	master	Signal	Power management outputs of cluster 30.
wake_request_31	master	Signal	Power management outputs of cluster 31.
wake_request_3	master	Signal	Power management outputs of cluster 3.
wake_request_4	master	Signal	Power management outputs of cluster 4.
wake_request_5	master	Signal	Power management outputs of cluster 5.
wake_request_6	master	Signal	Power management outputs of cluster 6.
wake_request_7	master	Signal	Power management outputs of cluster 7.
wake_request_8	master	Signal	Power management outputs of cluster 8.
wake_request_9	master	Signal	Power management outputs of cluster 9.

## Parameters for GIC500\_ClusterPorts

### **GICD\_ITARGETSR-RAZWI**

If true, the GICD\_ITARGETS registers are **RAZ/WI**.

Type: `bool`

Default value: `false`

### **ITS-count**

Number of Interrupt Translation Services to be instantiated (0=none).

Type: `uint8_t`

Default value: `1`

### **ITS-device-bits**

Number of bits supported for ITS device IDs.

Type: `int`

Default value: 16

### **ITS-threaded-command-queue**

Enable execution of ITS commands in a separate thread which is sometimes required for cosimulation.

Type: `bool`

Default value: `true`

### **SPI-count**

Number of SPIs that are implemented.

Type: `uint16_t`

Default value: 224

### **clear-ISPENDR-bit-for-level-sensitive-interrupt-when-acknowledged**

If true, acknowledgement of level sensitive interrupt clears the ISPENDR register.

Type: `bool`

Default value: `false`

### **cpus\_per\_cluster\_0**

Number of cores within cluster 0.

Type: `uint8_t`

Default value: 1

### **cpus\_per\_cluster\_1**

Number of cores within cluster 1.

Type: `uint8_t`

Default value: 1

### **cpus\_per\_cluster\_10**

Number of cores within cluster 10.

Type: `uint8_t`

Default value: 1

### **cpus\_per\_cluster\_11**

Number of cores within cluster 11.

Type: `uint8_t`

Default value: 1

### **`cpus_per_cluster_12`**

Number of cores within cluster 12.

Type: `uint8_t`

Default value: 1

### **`cpus_per_cluster_13`**

Number of cores within cluster 13.

Type: `uint8_t`

Default value: 1

### **`cpus_per_cluster_14`**

Number of cores within cluster 14.

Type: `uint8_t`

Default value: 1

### **`cpus_per_cluster_15`**

Number of cores within cluster 15.

Type: `uint8_t`

Default value: 1

### **`cpus_per_cluster_16`**

Number of cores within cluster 16.

Type: `uint8_t`

Default value: 1

### **`cpus_per_cluster_17`**

Number of cores within cluster 17.

Type: `uint8_t`

Default value: 1

### **`cpus_per_cluster_18`**

Number of cores within cluster 18.

Type: `uint8_t`

Default value: 1

### **`cpus_per_cluster_19`**

Number of cores within cluster 19.

Type: `uint8_t`

Default value: 1

### **`cpus_per_cluster_2`**

Number of cores within cluster 2.

Type: `uint8_t`

Default value: 1

### **`cpus_per_cluster_20`**

Number of cores within cluster 20.

Type: `uint8_t`

Default value: 1

### **`cpus_per_cluster_21`**

Number of cores within cluster 21.

Type: `uint8_t`

Default value: 1

### **`cpus_per_cluster_22`**

Number of cores within cluster 22.

Type: `uint8_t`

Default value: 1

### **`cpus_per_cluster_23`**

Number of cores within cluster 23.

Type: `uint8_t`

Default value: 1

### **`cpus_per_cluster_24`**

Number of cores within cluster 24.

Type: `uint8_t`

Default value: 1

### **`cpus_per_cluster_25`**

Number of cores within cluster 25.

Type: `uint8_t`

Default value: 1

### **`cpus_per_cluster_26`**

Number of cores within cluster 26.

Type: `uint8_t`

Default value: 1

### **`cpus_per_cluster_27`**

Number of cores within cluster 27.

Type: `uint8_t`

Default value: 1

### **`cpus_per_cluster_28`**

Number of cores within cluster 28.

Type: `uint8_t`

Default value: 1

### **`cpus_per_cluster_29`**

Number of cores within cluster 29.

Type: `uint8_t`

Default value: 1

### **`cpus_per_cluster_3`**

Number of cores within cluster 3.

Type: `uint8_t`

Default value: 1

### **`cpus_per_cluster_30`**

Number of cores within cluster 30.

Type: `uint8_t`

Default value: 1

### **`cpus_per_cluster_31`**

Number of cores within cluster 31.

Type: `uint8_t`

Default value: 1

### **`cpus_per_cluster_4`**

Number of cores within cluster 4.

Type: `uint8_t`

Default value: 1

### **`cpus_per_cluster_5`**

Number of cores within cluster 5.

Type: `uint8_t`

Default value: 1

### **`cpus_per_cluster_6`**

Number of cores within cluster 6.

Type: `uint8_t`

Default value: 1

### **`cpus_per_cluster_7`**

Number of cores within cluster 7.

Type: `uint8_t`

Default value: 1

### **`cpus_per_cluster_8`**

Number of cores within cluster 8.

Type: `uint8_t`

Default value: 1

### **`cpus_per_cluster_9`**

Number of cores within cluster 9.

Type: `uint8_t`

Default value: `1`

### **delay-ITS-accesses**

Delay accesses from the ITS until GICR\_SYNCNR is read.

Type: `bool`

Default value: `true`

### **delay-redistributor-accesses**

Delay memory accesses from the redistributor until GICR\_SYNCNR is read.

Type: `bool`

Default value: `true`

### **enable\_protocol\_checking**

Enable/disable protocol checking at cpu interface.

Type: `bool`

Default value: `false`

### **enabled**

Enable GICv3 functionality; when false the component is inactive.

Type: `bool`

Default value: `true`

### **has-two-security-states**

If true, has two security states.

Type: `bool`

Default value: `true`

### **num\_clusters**

Number of implemented affinity level1 clusters.

Type: `uint8_t`

Default value: `1`

### **print-memory-map**

Print memory map to stdout.



Type: `bool`

Default value: `false`

### **redistributor-threaded-sync**

Enable execution of redistributor delayed transactions in a separate thread which is sometimes required for cosimulation.

Type: `bool`

Default value: `true`

### **reg-base**

GIC500 base address.

Type: `uint64_t`

Default value: `0x2c010000`

### **using-generated-memorymap**

Use the generated memorymap for the GIC500 and warn if superfluous parameters are passed.

Type: `bool`

Default value: `true`

### **wakeup-on-reset**

Go against specification and start redistributors in woken-up state at reset. This allows software that was written for previous versions of the GICv3 specification to work correctly. This should not be used for production code or when the distributor is used separately from the core fast model.

Type: `bool`

Default value: `false`

## 3.187 GIC500\_Filter

Defined in `LISA/GIC500_Filter.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

## About GIC500\_Filter

This is a single-component implementation of the GICv3 architecture with support for 256 cores. You can configure the model to support a maximum of 32 clusters with 8 cores per cluster. Use it with an Armv8-A core to deliver interrupts. It supports a single Interrupt Translation Service for message-based interrupts. It supports the architectural features, but does not support the implementation defined features.



Note

- Set the `FASTSIM_GIC_MEMORY_MAP` environment variable to 1 to print to stderr the memory map of any GICv3 or later models that are included in the platform being run.
- For writes to `GITS_TRANSLATE64R`, the `MSIRewriter` component can be used to create the correct data transactions based on the device ID. For more details about `GITS_TRANSLATE64R`, see [MSIRewriter](#).

## ManagerID, ExtendedID, and UserFlags

This table shows how this model encodes information in the `ManagerID`, `ExtendedID`, and `UserFlags` bus attributes:

**Table 3-644: ManagerID, ExtendedID, and UserFlags**

PVBus attribute	Bits used	Property encoded	Notes
ManagerID	63:0	Manager ID that invoked the register access	–
ExtendedID	–	–	Not used
UserFlags	–	–	Not used



Note

The `pvbus_m` port does not use `ManagerID`, `ExtendedID`, or `UserFlags`.

## Iris and MTI instances for GIC500\_Filter

This model has the following Iris instances:

Name	Instance type
GIC500_Filter	GIC500
GIC500_Filter.ITS0	GICv3InterruptTranslationService
GIC500_Filter.ITS0.ExportTest.GIC500_Filter.ITS0.pvbus_m[0].pvbusmaster	PVBusMaster
GIC500_Filter.rd_0	GICv3RedistributorInternal
GIC500_Filter.rd_0_0	GICv3RedistributorInternal
GIC500_Filter.rd_0_0_0	GICv3RedistributorInternal
GIC500_Filter.rd_0_0_0_0	GICv3Redistributor
GIC500_Filter.rd_0_0_0_0.ExportTest.GIC500_Filter.rd_0_0_0_0.pvbus_m[0].pvbusmaster	PVBusMaster
GIC500_Filter.rd_t1	GICv3Distributor

This model has the following MTI trace components:

Name	Component type
GIC500_Filter.ITS0	GICv3InterruptTranslationService
GIC500_Filter.ITS0.ExportTest.GIC500_Filter.ITS0.pvbus_m[0].pvbusmaster	PVBusMaster
GIC500_Filter.rd_0	GICv3RedistributorInternal
GIC500_Filter.rd_0_0	GICv3RedistributorInternal
GIC500_Filter.rd_0_0_0	GICv3RedistributorInternal
GIC500_Filter.rd_0_0_0_0	GICv3Redistributor
GIC500_Filter.rd_0_0_0_0.ExportTest.GIC500_Filter.rd_0_0_0_0.pvbus_m[0].pvbusmaster	PVBusMaster
GIC500_Filter.rd_tl	GICv3Distributor

### Ports for GIC500\_Filter

Port	Direction	Protocol	Description
cfgsdisable	slave	Signal	Disable some SPLs signal.
cpu_active_0	slave	Signal	cpu_active pins of cluster 0.
cpu_active_10	slave	Signal	cpu_active pins of cluster 10.
cpu_active_11	slave	Signal	cpu_active pins of cluster 11.
cpu_active_12	slave	Signal	cpu_active pins of cluster 12.
cpu_active_13	slave	Signal	cpu_active pins of cluster 13.
cpu_active_14	slave	Signal	cpu_active pins of cluster 14.
cpu_active_15	slave	Signal	cpu_active pins of cluster 15.
cpu_active_16	slave	Signal	cpu_active pins of cluster 16.
cpu_active_17	slave	Signal	cpu_active pins of cluster 17.
cpu_active_18	slave	Signal	cpu_active pins of cluster 18.
cpu_active_19	slave	Signal	cpu_active pins of cluster 19.
cpu_active_1	slave	Signal	cpu_active pins of cluster 1.
cpu_active_20	slave	Signal	cpu_active pins of cluster 20.
cpu_active_21	slave	Signal	cpu_active pins of cluster 21.
cpu_active_22	slave	Signal	cpu_active pins of cluster 22.
cpu_active_23	slave	Signal	cpu_active pins of cluster 23.
cpu_active_24	slave	Signal	cpu_active pins of cluster 24.
cpu_active_25	slave	Signal	cpu_active pins of cluster 25.
cpu_active_26	slave	Signal	cpu_active pins of cluster 26.
cpu_active_27	slave	Signal	cpu_active pins of cluster 27.
cpu_active_28	slave	Signal	cpu_active pins of cluster 28.
cpu_active_29	slave	Signal	cpu_active pins of cluster 29.
cpu_active_2	slave	Signal	cpu_active pins of cluster 2.
cpu_active_30	slave	Signal	cpu_active pins of cluster 30.
cpu_active_31	slave	Signal	cpu_active pins of cluster 31.
cpu_active_3	slave	Signal	cpu_active pins of cluster 3.

Port	Direction	Protocol	Description
cpu_active_4	slave	Signal	cpu_active pins of cluster 4.
cpu_active_5	slave	Signal	cpu_active pins of cluster 5.
cpu_active_6	slave	Signal	cpu_active pins of cluster 6.
cpu_active_7	slave	Signal	cpu_active pins of cluster 7.
cpu_active_8	slave	Signal	cpu_active pins of cluster 8.
cpu_active_9	slave	Signal	cpu_active pins of cluster 9.
po_reset	slave	Signal	Power on reset.
ppi16_in_0	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 0.
ppi16_in_10	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 10.
ppi16_in_11	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 11.
ppi16_in_12	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 12.
ppi16_in_13	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 13.
ppi16_in_14	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 14.
ppi16_in_15	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 15.
ppi16_in_16	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 16.
ppi16_in_17	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 17.
ppi16_in_18	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 18.
ppi16_in_19	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 19.
ppi16_in_1	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 1.
ppi16_in_20	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 20.
ppi16_in_21	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 21.
ppi16_in_22	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 22.
ppi16_in_23	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 23.
ppi16_in_24	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 24.
ppi16_in_25	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 25.
ppi16_in_26	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 26.
ppi16_in_27	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 27.
ppi16_in_28	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 28.
ppi16_in_29	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 29.
ppi16_in_2	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 2.
ppi16_in_30	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 30.
ppi16_in_31	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 31.
ppi16_in_3	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 3.
ppi16_in_4	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 4.
ppi16_in_5	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 5.
ppi16_in_6	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 6.
ppi16_in_7	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 7.
ppi16_in_8	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 8.
ppi16_in_9	slave	Signal	Private peripheral interrupts interrupts (ID16) of cluster 9.
ppi17_in_0	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 0.

Port	Direction	Protocol	Description
ppi17_in_10	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 10.
ppi17_in_11	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 11.
ppi17_in_12	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 12.
ppi17_in_13	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 13.
ppi17_in_14	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 14.
ppi17_in_15	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 15.
ppi17_in_16	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 16.
ppi17_in_17	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 17.
ppi17_in_18	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 18.
ppi17_in_19	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 19.
ppi17_in_1	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 1.
ppi17_in_20	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 20.
ppi17_in_21	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 21.
ppi17_in_22	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 22.
ppi17_in_23	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 23.
ppi17_in_24	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 24.
ppi17_in_25	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 25.
ppi17_in_26	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 26.
ppi17_in_27	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 27.
ppi17_in_28	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 28.
ppi17_in_29	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 29.
ppi17_in_2	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 2.
ppi17_in_30	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 30.
ppi17_in_31	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 31.
ppi17_in_3	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 3.
ppi17_in_4	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 4.
ppi17_in_5	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 5.
ppi17_in_6	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 6.
ppi17_in_7	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 7.
ppi17_in_8	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 8.
ppi17_in_9	slave	Signal	Private peripheral interrupts interrupts (ID17) of cluster 9.
ppi18_in_0	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 0.
ppi18_in_10	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 10.
ppi18_in_11	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 11.
ppi18_in_12	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 12.
ppi18_in_13	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 13.
ppi18_in_14	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 14.
ppi18_in_15	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 15.
ppi18_in_16	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 16.
ppi18_in_17	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 17.

Port	Direction	Protocol	Description
ppi18_in_18	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 18.
ppi18_in_19	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 19.
ppi18_in_1	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 1.
ppi18_in_20	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 20.
ppi18_in_21	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 21.
ppi18_in_22	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 22.
ppi18_in_23	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 23.
ppi18_in_24	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 24.
ppi18_in_25	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 25.
ppi18_in_26	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 26.
ppi18_in_27	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 27.
ppi18_in_28	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 28.
ppi18_in_29	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 29.
ppi18_in_2	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 2.
ppi18_in_30	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 30.
ppi18_in_31	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 31.
ppi18_in_3	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 3.
ppi18_in_4	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 4.
ppi18_in_5	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 5.
ppi18_in_6	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 6.
ppi18_in_7	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 7.
ppi18_in_8	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 8.
ppi18_in_9	slave	Signal	Private peripheral interrupts interrupts (ID18) of cluster 9.
ppi19_in_0	slave	Signal	Private peripheral interrupts (ID19) of cluster 0.
ppi19_in_10	slave	Signal	Private peripheral interrupts (ID19) of cluster 10.
ppi19_in_11	slave	Signal	Private peripheral interrupts (ID19) of cluster 11.
ppi19_in_12	slave	Signal	Private peripheral interrupts (ID19) of cluster 12.
ppi19_in_13	slave	Signal	Private peripheral interrupts (ID19) of cluster 13.
ppi19_in_14	slave	Signal	Private peripheral interrupts (ID19) of cluster 14.
ppi19_in_15	slave	Signal	Private peripheral interrupts (ID19) of cluster 15.
ppi19_in_16	slave	Signal	Private peripheral interrupts (ID19) of cluster 16.
ppi19_in_17	slave	Signal	Private peripheral interrupts (ID19) of cluster 17.
ppi19_in_18	slave	Signal	Private peripheral interrupts (ID19) of cluster 18.
ppi19_in_19	slave	Signal	Private peripheral interrupts (ID19) of cluster 19.
ppi19_in_1	slave	Signal	Private peripheral interrupts (ID19) of cluster 1.
ppi19_in_20	slave	Signal	Private peripheral interrupts (ID19) of cluster 20.
ppi19_in_21	slave	Signal	Private peripheral interrupts (ID19) of cluster 21.
ppi19_in_22	slave	Signal	Private peripheral interrupts (ID19) of cluster 22.
ppi19_in_23	slave	Signal	Private peripheral interrupts (ID19) of cluster 23.
ppi19_in_24	slave	Signal	Private peripheral interrupts (ID19) of cluster 24.

Port	Direction	Protocol	Description
ppi19_in_25	slave	Signal	Private peripheral interrupts (ID19) of cluster 25.
ppi19_in_26	slave	Signal	Private peripheral interrupts (ID19) of cluster 26.
ppi19_in_27	slave	Signal	Private peripheral interrupts (ID19) of cluster 27.
ppi19_in_28	slave	Signal	Private peripheral interrupts (ID19) of cluster 28.
ppi19_in_29	slave	Signal	Private peripheral interrupts (ID19) of cluster 29.
ppi19_in_2	slave	Signal	Private peripheral interrupts (ID19) of cluster 2.
ppi19_in_30	slave	Signal	Private peripheral interrupts (ID19) of cluster 30.
ppi19_in_31	slave	Signal	Private peripheral interrupts (ID19) of cluster 31.
ppi19_in_3	slave	Signal	Private peripheral interrupts (ID19) of cluster 3.
ppi19_in_4	slave	Signal	Private peripheral interrupts (ID19) of cluster 4.
ppi19_in_5	slave	Signal	Private peripheral interrupts (ID19) of cluster 5.
ppi19_in_6	slave	Signal	Private peripheral interrupts (ID19) of cluster 6.
ppi19_in_7	slave	Signal	Private peripheral interrupts (ID19) of cluster 7.
ppi19_in_8	slave	Signal	Private peripheral interrupts (ID19) of cluster 8.
ppi19_in_9	slave	Signal	Private peripheral interrupts (ID19) of cluster 9.
ppi20_in_0	slave	Signal	Private peripheral interrupts (ID20) of cluster 0.
ppi20_in_10	slave	Signal	Private peripheral interrupts (ID20) of cluster 10.
ppi20_in_11	slave	Signal	Private peripheral interrupts (ID20) of cluster 11.
ppi20_in_12	slave	Signal	Private peripheral interrupts (ID20) of cluster 12.
ppi20_in_13	slave	Signal	Private peripheral interrupts (ID20) of cluster 13.
ppi20_in_14	slave	Signal	Private peripheral interrupts (ID20) of cluster 14.
ppi20_in_15	slave	Signal	Private peripheral interrupts (ID20) of cluster 15.
ppi20_in_16	slave	Signal	Private peripheral interrupts (ID20) of cluster 16.
ppi20_in_17	slave	Signal	Private peripheral interrupts (ID20) of cluster 17.
ppi20_in_18	slave	Signal	Private peripheral interrupts (ID20) of cluster 18.
ppi20_in_19	slave	Signal	Private peripheral interrupts (ID20) of cluster 19.
ppi20_in_1	slave	Signal	Private peripheral interrupts (ID20) of cluster 1.
ppi20_in_20	slave	Signal	Private peripheral interrupts (ID20) of cluster 20.
ppi20_in_21	slave	Signal	Private peripheral interrupts (ID20) of cluster 21.
ppi20_in_22	slave	Signal	Private peripheral interrupts (ID20) of cluster 22.
ppi20_in_23	slave	Signal	Private peripheral interrupts (ID20) of cluster 23.
ppi20_in_24	slave	Signal	Private peripheral interrupts (ID20) of cluster 24.
ppi20_in_25	slave	Signal	Private peripheral interrupts (ID20) of cluster 25.
ppi20_in_26	slave	Signal	Private peripheral interrupts (ID20) of cluster 26.
ppi20_in_27	slave	Signal	Private peripheral interrupts (ID20) of cluster 27.
ppi20_in_28	slave	Signal	Private peripheral interrupts (ID20) of cluster 28.
ppi20_in_29	slave	Signal	Private peripheral interrupts (ID20) of cluster 29.
ppi20_in_2	slave	Signal	Private peripheral interrupts (ID20) of cluster 2.
ppi20_in_30	slave	Signal	Private peripheral interrupts (ID20) of cluster 30.
ppi20_in_31	slave	Signal	Private peripheral interrupts (ID20) of cluster 31.

Port	Direction	Protocol	Description
ppi20_in_3	slave	Signal	Private peripheral interrupts (ID20) of cluster 3.
ppi20_in_4	slave	Signal	Private peripheral interrupts (ID20) of cluster 4.
ppi20_in_5	slave	Signal	Private peripheral interrupts (ID20) of cluster 5.
ppi20_in_6	slave	Signal	Private peripheral interrupts (ID20) of cluster 6.
ppi20_in_7	slave	Signal	Private peripheral interrupts (ID20) of cluster 7.
ppi20_in_8	slave	Signal	Private peripheral interrupts (ID20) of cluster 8.
ppi20_in_9	slave	Signal	Private peripheral interrupts (ID20) of cluster 9.
ppi21_in_0	slave	Signal	Private peripheral interrupts (ID21) of cluster 0.
ppi21_in_10	slave	Signal	Private peripheral interrupts (ID21) of cluster 10.
ppi21_in_11	slave	Signal	Private peripheral interrupts (ID21) of cluster 11.
ppi21_in_12	slave	Signal	Private peripheral interrupts (ID21) of cluster 12.
ppi21_in_13	slave	Signal	Private peripheral interrupts (ID21) of cluster 13.
ppi21_in_14	slave	Signal	Private peripheral interrupts (ID21) of cluster 14.
ppi21_in_15	slave	Signal	Private peripheral interrupts (ID21) of cluster 15.
ppi21_in_16	slave	Signal	Private peripheral interrupts (ID21) of cluster 16.
ppi21_in_17	slave	Signal	Private peripheral interrupts (ID21) of cluster 17.
ppi21_in_18	slave	Signal	Private peripheral interrupts (ID21) of cluster 18.
ppi21_in_19	slave	Signal	Private peripheral interrupts (ID21) of cluster 19.
ppi21_in_1	slave	Signal	Private peripheral interrupts (ID21) of cluster 1.
ppi21_in_20	slave	Signal	Private peripheral interrupts (ID21) of cluster 20.
ppi21_in_21	slave	Signal	Private peripheral interrupts (ID21) of cluster 21.
ppi21_in_22	slave	Signal	Private peripheral interrupts (ID21) of cluster 22.
ppi21_in_23	slave	Signal	Private peripheral interrupts (ID21) of cluster 23.
ppi21_in_24	slave	Signal	Private peripheral interrupts (ID21) of cluster 24.
ppi21_in_25	slave	Signal	Private peripheral interrupts (ID21) of cluster 25.
ppi21_in_26	slave	Signal	Private peripheral interrupts (ID21) of cluster 26.
ppi21_in_27	slave	Signal	Private peripheral interrupts (ID21) of cluster 27.
ppi21_in_28	slave	Signal	Private peripheral interrupts (ID21) of cluster 28.
ppi21_in_29	slave	Signal	Private peripheral interrupts (ID21) of cluster 29.
ppi21_in_2	slave	Signal	Private peripheral interrupts (ID21) of cluster 2.
ppi21_in_30	slave	Signal	Private peripheral interrupts (ID21) of cluster 30.
ppi21_in_31	slave	Signal	Private peripheral interrupts (ID21) of cluster 31.
ppi21_in_3	slave	Signal	Private peripheral interrupts (ID21) of cluster 3.
ppi21_in_4	slave	Signal	Private peripheral interrupts (ID21) of cluster 4.
ppi21_in_5	slave	Signal	Private peripheral interrupts (ID21) of cluster 5.
ppi21_in_6	slave	Signal	Private peripheral interrupts (ID21) of cluster 6.
ppi21_in_7	slave	Signal	Private peripheral interrupts (ID21) of cluster 7.
ppi21_in_8	slave	Signal	Private peripheral interrupts (ID21) of cluster 8.
ppi21_in_9	slave	Signal	Private peripheral interrupts (ID21) of cluster 9.
ppi22_in_0	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 0.



Port	Direction	Protocol	Description
ppi22_in_10	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 10.
ppi22_in_11	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 11.
ppi22_in_12	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 12.
ppi22_in_13	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 13.
ppi22_in_14	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 14.
ppi22_in_15	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 15.
ppi22_in_16	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 16.
ppi22_in_17	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 17.
ppi22_in_18	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 18.
ppi22_in_19	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 19.
ppi22_in_1	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 1.
ppi22_in_20	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 20.
ppi22_in_21	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 21.
ppi22_in_22	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 22.
ppi22_in_23	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 23.
ppi22_in_24	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 24.
ppi22_in_25	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 25.
ppi22_in_26	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 26.
ppi22_in_27	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 27.
ppi22_in_28	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 28.
ppi22_in_29	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 29.
ppi22_in_2	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 2.
ppi22_in_30	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 30.

Port	Direction	Protocol	Description
ppi22_in_31	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 31.
ppi22_in_3	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 3.
ppi22_in_4	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 4.
ppi22_in_5	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 5.
ppi22_in_6	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 6.
ppi22_in_7	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 7.
ppi22_in_8	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 8.
ppi22_in_9	slave	Signal	Private peripheral interrupts (ID22, typically the Debug Communications Channel (DCC) interrupt) of cluster 9.
ppi23_in_0	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 0.
ppi23_in_10	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 10.
ppi23_in_11	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 11.
ppi23_in_12	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 12.
ppi23_in_13	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 13.
ppi23_in_14	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 14.
ppi23_in_15	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 15.
ppi23_in_16	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 16.
ppi23_in_17	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 17.
ppi23_in_18	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 18.
ppi23_in_19	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 19.
ppi23_in_1	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 1.
ppi23_in_20	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 20.
ppi23_in_21	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 21.
ppi23_in_22	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 22.

Port	Direction	Protocol	Description
ppi23_in_23	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 23.
ppi23_in_24	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 24.
ppi23_in_25	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 25.
ppi23_in_26	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 26.
ppi23_in_27	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 27.
ppi23_in_28	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 28.
ppi23_in_29	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 29.
ppi23_in_2	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 2.
ppi23_in_30	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 30.
ppi23_in_31	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 31.
ppi23_in_3	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 3.
ppi23_in_4	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 4.
ppi23_in_5	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 5.
ppi23_in_6	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 6.
ppi23_in_7	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 7.
ppi23_in_8	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 8.
ppi23_in_9	slave	Signal	Private peripheral interrupts (ID23, typically the Performance Counter (PMU) overflow interrupt) of cluster 9.
ppi24_in_0	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 0.
ppi24_in_10	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 10.
ppi24_in_11	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 11.
ppi24_in_12	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 12.
ppi24_in_13	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 13.
ppi24_in_14	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 14.

Port	Direction	Protocol	Description
ppi24_in_15	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 15.
ppi24_in_16	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 16.
ppi24_in_17	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 17.
ppi24_in_18	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 18.
ppi24_in_19	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 19.
ppi24_in_1	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 1.
ppi24_in_20	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 20.
ppi24_in_21	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 21.
ppi24_in_22	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 22.
ppi24_in_23	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 23.
ppi24_in_24	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 24.
ppi24_in_25	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 25.
ppi24_in_26	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 26.
ppi24_in_27	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 27.
ppi24_in_28	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 28.
ppi24_in_29	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 29.
ppi24_in_2	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 2.
ppi24_in_30	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 30.
ppi24_in_31	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 31.
ppi24_in_3	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 3.
ppi24_in_4	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 4.
ppi24_in_5	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 5.
ppi24_in_6	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 6.

Port	Direction	Protocol	Description
ppi24_in_7	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 7.
ppi24_in_8	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 8.
ppi24_in_9	slave	Signal	Private peripheral interrupts (ID24, typically the Cross Trigger Interface (CTI) interrupt) of cluster 9.
ppi25_in_0	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 0.
ppi25_in_10	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 10.
ppi25_in_11	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 11.
ppi25_in_12	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 12.
ppi25_in_13	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 13.
ppi25_in_14	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 14.
ppi25_in_15	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 15.
ppi25_in_16	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 16.
ppi25_in_17	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 17.
ppi25_in_18	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 18.
ppi25_in_19	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 19.
ppi25_in_1	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 1.
ppi25_in_20	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 20.
ppi25_in_21	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 21.
ppi25_in_22	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 22.
ppi25_in_23	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 23.
ppi25_in_24	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 24.
ppi25_in_25	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 25.
ppi25_in_26	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 26.
ppi25_in_27	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 27.

Port	Direction	Protocol	Description
ppi25_in_28	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 28.
ppi25_in_29	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 29.
ppi25_in_2	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 2.
ppi25_in_30	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 30.
ppi25_in_31	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 31.
ppi25_in_3	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 3.
ppi25_in_4	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 4.
ppi25_in_5	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 5.
ppi25_in_6	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 6.
ppi25_in_7	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 7.
ppi25_in_8	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 8.
ppi25_in_9	slave	Signal	Private peripheral interrupts (ID25, typically the virtual CPU interface maintenance interrupt) of cluster 9.
ppi26_in_0	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 0.
ppi26_in_10	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 10.
ppi26_in_11	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 11.
ppi26_in_12	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 12.
ppi26_in_13	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 13.
ppi26_in_14	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 14.
ppi26_in_15	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 15.
ppi26_in_16	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 16.
ppi26_in_17	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 17.
ppi26_in_18	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 18.
ppi26_in_19	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 19.
ppi26_in_1	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 1.
ppi26_in_20	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 20.
ppi26_in_21	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 21.
ppi26_in_22	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 22.
ppi26_in_23	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 23.
ppi26_in_24	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 24.
ppi26_in_25	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 25.
ppi26_in_26	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 26.
ppi26_in_27	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 27.

Port	Direction	Protocol	Description
ppi26_in_28	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 28.
ppi26_in_29	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 29.
ppi26_in_2	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 2.
ppi26_in_30	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 30.
ppi26_in_31	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 31.
ppi26_in_3	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 3.
ppi26_in_4	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 4.
ppi26_in_5	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 5.
ppi26_in_6	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 6.
ppi26_in_7	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 7.
ppi26_in_8	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 8.
ppi26_in_9	slave	Signal	Private peripheral interrupts (ID26, typically the hypervisor timer) of cluster 9.
ppi27_in_0	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 0.
ppi27_in_10	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 10.
ppi27_in_11	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 11.
ppi27_in_12	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 12.
ppi27_in_13	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 13.
ppi27_in_14	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 14.
ppi27_in_15	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 15.
ppi27_in_16	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 16.
ppi27_in_17	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 17.
ppi27_in_18	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 18.
ppi27_in_19	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 19.
ppi27_in_1	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 1.
ppi27_in_20	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 20.
ppi27_in_21	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 21.
ppi27_in_22	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 22.
ppi27_in_23	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 23.
ppi27_in_24	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 24.
ppi27_in_25	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 25.
ppi27_in_26	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 26.
ppi27_in_27	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 27.
ppi27_in_28	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 28.
ppi27_in_29	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 29.
ppi27_in_2	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 2.
ppi27_in_30	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 30.
ppi27_in_31	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 31.
ppi27_in_3	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 3.
ppi27_in_4	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 4.
ppi27_in_5	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 5.



Port	Direction	Protocol	Description
ppi27_in_6	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 6.
ppi27_in_7	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 7.
ppi27_in_8	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 8.
ppi27_in_9	slave	Signal	Private peripheral interrupts (ID27, typically the virtual timer) of cluster 9.
ppi28_in_0	slave	Signal	Private peripheral interrupts (ID28) of cluster 0.
ppi28_in_10	slave	Signal	Private peripheral interrupts (ID28) of cluster 10.
ppi28_in_11	slave	Signal	Private peripheral interrupts (ID28) of cluster 11.
ppi28_in_12	slave	Signal	Private peripheral interrupts (ID28) of cluster 12.
ppi28_in_13	slave	Signal	Private peripheral interrupts (ID28) of cluster 13.
ppi28_in_14	slave	Signal	Private peripheral interrupts (ID28) of cluster 14.
ppi28_in_15	slave	Signal	Private peripheral interrupts (ID28) of cluster 15.
ppi28_in_16	slave	Signal	Private peripheral interrupts (ID28) of cluster 16.
ppi28_in_17	slave	Signal	Private peripheral interrupts (ID28) of cluster 17.
ppi28_in_18	slave	Signal	Private peripheral interrupts (ID28) of cluster 18.
ppi28_in_19	slave	Signal	Private peripheral interrupts (ID28) of cluster 19.
ppi28_in_1	slave	Signal	Private peripheral interrupts (ID28) of cluster 1.
ppi28_in_20	slave	Signal	Private peripheral interrupts (ID28) of cluster 20.
ppi28_in_21	slave	Signal	Private peripheral interrupts (ID28) of cluster 21.
ppi28_in_22	slave	Signal	Private peripheral interrupts (ID28) of cluster 22.
ppi28_in_23	slave	Signal	Private peripheral interrupts (ID28) of cluster 23.
ppi28_in_24	slave	Signal	Private peripheral interrupts (ID28) of cluster 24.
ppi28_in_25	slave	Signal	Private peripheral interrupts (ID28) of cluster 25.
ppi28_in_26	slave	Signal	Private peripheral interrupts (ID28) of cluster 26.
ppi28_in_27	slave	Signal	Private peripheral interrupts (ID28) of cluster 27.
ppi28_in_28	slave	Signal	Private peripheral interrupts (ID28) of cluster 28.
ppi28_in_29	slave	Signal	Private peripheral interrupts (ID28) of cluster 29.
ppi28_in_2	slave	Signal	Private peripheral interrupts (ID28) of cluster 2.
ppi28_in_30	slave	Signal	Private peripheral interrupts (ID28) of cluster 30.
ppi28_in_31	slave	Signal	Private peripheral interrupts (ID28) of cluster 31.
ppi28_in_3	slave	Signal	Private peripheral interrupts (ID28) of cluster 3.
ppi28_in_4	slave	Signal	Private peripheral interrupts (ID28) of cluster 4.
ppi28_in_5	slave	Signal	Private peripheral interrupts (ID28) of cluster 5.
ppi28_in_6	slave	Signal	Private peripheral interrupts (ID28) of cluster 6.
ppi28_in_7	slave	Signal	Private peripheral interrupts (ID28) of cluster 7.
ppi28_in_8	slave	Signal	Private peripheral interrupts (ID28) of cluster 8.
ppi28_in_9	slave	Signal	Private peripheral interrupts (ID28) of cluster 9.
ppi29_in_0	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 0.
ppi29_in_10	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 10.
ppi29_in_11	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 11.
ppi29_in_12	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 12.



Port	Direction	Protocol	Description
ppi29_in_13	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 13.
ppi29_in_14	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 14.
ppi29_in_15	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 15.
ppi29_in_16	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 16.
ppi29_in_17	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 17.
ppi29_in_18	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 18.
ppi29_in_19	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 19.
ppi29_in_1	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 1.
ppi29_in_20	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 20.
ppi29_in_21	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 21.
ppi29_in_22	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 22.
ppi29_in_23	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 23.
ppi29_in_24	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 24.
ppi29_in_25	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 25.
ppi29_in_26	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 26.
ppi29_in_27	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 27.
ppi29_in_28	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 28.
ppi29_in_29	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 29.
ppi29_in_2	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 2.
ppi29_in_30	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 30.
ppi29_in_31	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 31.
ppi29_in_3	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 3.
ppi29_in_4	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 4.
ppi29_in_5	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 5.
ppi29_in_6	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 6.
ppi29_in_7	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 7.
ppi29_in_8	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 8.
ppi29_in_9	slave	Signal	Private peripheral interrupts (ID29, typically the Secure physical timer) of cluster 9.
ppi30_in_0	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 0.
ppi30_in_10	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 10.
ppi30_in_11	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 11.
ppi30_in_12	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 12.
ppi30_in_13	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 13.
ppi30_in_14	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 14.
ppi30_in_15	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 15.

Port	Direction	Protocol	Description
ppi30_in_16	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 16.
ppi30_in_17	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 17.
ppi30_in_18	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 18.
ppi30_in_19	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 19.
ppi30_in_1	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 1.
ppi30_in_20	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 20.
ppi30_in_21	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 21.
ppi30_in_22	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 22.
ppi30_in_23	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 23.
ppi30_in_24	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 24.
ppi30_in_25	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 25.
ppi30_in_26	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 26.
ppi30_in_27	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 27.
ppi30_in_28	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 28.
ppi30_in_29	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 29.
ppi30_in_2	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 2.
ppi30_in_30	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 30.
ppi30_in_31	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 31.
ppi30_in_3	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 3.
ppi30_in_4	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 4.
ppi30_in_5	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 5.
ppi30_in_6	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 6.
ppi30_in_7	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 7.

Port	Direction	Protocol	Description
ppi30_in_8	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 8.
ppi30_in_9	slave	Signal	Private peripheral interrupts (ID30, typically the Non-secure physical timer) of cluster 9.
ppi31_in_0	slave	Signal	Private peripheral interrupts (ID31) of cluster 0.
ppi31_in_10	slave	Signal	Private peripheral interrupts (ID31) of cluster 10.
ppi31_in_11	slave	Signal	Private peripheral interrupts (ID31) of cluster 11.
ppi31_in_12	slave	Signal	Private peripheral interrupts (ID31) of cluster 12.
ppi31_in_13	slave	Signal	Private peripheral interrupts (ID31) of cluster 13.
ppi31_in_14	slave	Signal	Private peripheral interrupts (ID31) of cluster 14.
ppi31_in_15	slave	Signal	Private peripheral interrupts (ID31) of cluster 15.
ppi31_in_16	slave	Signal	Private peripheral interrupts (ID31) of cluster 16.
ppi31_in_17	slave	Signal	Private peripheral interrupts (ID31) of cluster 17.
ppi31_in_18	slave	Signal	Private peripheral interrupts (ID31) of cluster 18.
ppi31_in_19	slave	Signal	Private peripheral interrupts (ID31) of cluster 19.
ppi31_in_1	slave	Signal	Private peripheral interrupts (ID31) of cluster 1.
ppi31_in_20	slave	Signal	Private peripheral interrupts (ID31) of cluster 20.
ppi31_in_21	slave	Signal	Private peripheral interrupts (ID31) of cluster 21.
ppi31_in_22	slave	Signal	Private peripheral interrupts (ID31) of cluster 22.
ppi31_in_23	slave	Signal	Private peripheral interrupts (ID31) of cluster 23.
ppi31_in_24	slave	Signal	Private peripheral interrupts (ID31) of cluster 24.
ppi31_in_25	slave	Signal	Private peripheral interrupts (ID31) of cluster 25.
ppi31_in_26	slave	Signal	Private peripheral interrupts (ID31) of cluster 26.
ppi31_in_27	slave	Signal	Private peripheral interrupts (ID31) of cluster 27.
ppi31_in_28	slave	Signal	Private peripheral interrupts (ID31) of cluster 28.
ppi31_in_29	slave	Signal	Private peripheral interrupts (ID31) of cluster 29.
ppi31_in_2	slave	Signal	Private peripheral interrupts (ID31) of cluster 2.
ppi31_in_30	slave	Signal	Private peripheral interrupts (ID31) of cluster 30.
ppi31_in_31	slave	Signal	Private peripheral interrupts (ID31) of cluster 31.
ppi31_in_3	slave	Signal	Private peripheral interrupts (ID31) of cluster 3.
ppi31_in_4	slave	Signal	Private peripheral interrupts (ID31) of cluster 4.
ppi31_in_5	slave	Signal	Private peripheral interrupts (ID31) of cluster 5.
ppi31_in_6	slave	Signal	Private peripheral interrupts (ID31) of cluster 6.
ppi31_in_7	slave	Signal	Private peripheral interrupts (ID31) of cluster 7.
ppi31_in_8	slave	Signal	Private peripheral interrupts (ID31) of cluster 8.
ppi31_in_9	slave	Signal	Private peripheral interrupts (ID31) of cluster 9.
pvbus_filtermiss_m	master	PVBus	passthrough for transactions not targetting one of the pages associated with the IRI.
pvbus_m	master	PVBus	Memory bus out: transactions generated by the IRI.
pvbus_s	slave	PVBus	Memory bus in: memory-mapped register accesses are accepted on this interface.
redistributor_m	master	GLCv3Comms	Input from and output to CPU interface.
reset	slave	Signal	Reset.

Port	Direction	Protocol	Description
<code>spi_in</code>	slave	Signal	Shared peripheral interrupts.
<code>wake_request_0</code>	master	Signal	Power management outputs of cluster 0.
<code>wake_request_10</code>	master	Signal	Power management outputs of cluster 10.
<code>wake_request_11</code>	master	Signal	Power management outputs of cluster 11.
<code>wake_request_12</code>	master	Signal	Power management outputs of cluster 12.
<code>wake_request_13</code>	master	Signal	Power management outputs of cluster 13.
<code>wake_request_14</code>	master	Signal	Power management outputs of cluster 14.
<code>wake_request_15</code>	master	Signal	Power management outputs of cluster 15.
<code>wake_request_16</code>	master	Signal	Power management outputs of cluster 16.
<code>wake_request_17</code>	master	Signal	Power management outputs of cluster 17.
<code>wake_request_18</code>	master	Signal	Power management outputs of cluster 18.
<code>wake_request_19</code>	master	Signal	Power management outputs of cluster 19.
<code>wake_request_1</code>	master	Signal	Power management outputs of cluster 1.
<code>wake_request_20</code>	master	Signal	Power management outputs of cluster 20.
<code>wake_request_21</code>	master	Signal	Power management outputs of cluster 21.
<code>wake_request_22</code>	master	Signal	Power management outputs of cluster 22.
<code>wake_request_23</code>	master	Signal	Power management outputs of cluster 23.
<code>wake_request_24</code>	master	Signal	Power management outputs of cluster 24.
<code>wake_request_25</code>	master	Signal	Power management outputs of cluster 25.
<code>wake_request_26</code>	master	Signal	Power management outputs of cluster 26.
<code>wake_request_27</code>	master	Signal	Power management outputs of cluster 27.
<code>wake_request_28</code>	master	Signal	Power management outputs of cluster 28.
<code>wake_request_29</code>	master	Signal	Power management outputs of cluster 29.
<code>wake_request_2</code>	master	Signal	Power management outputs of cluster 2.
<code>wake_request_30</code>	master	Signal	Power management outputs of cluster 30.
<code>wake_request_31</code>	master	Signal	Power management outputs of cluster 31.
<code>wake_request_3</code>	master	Signal	Power management outputs of cluster 3.
<code>wake_request_4</code>	master	Signal	Power management outputs of cluster 4.
<code>wake_request_5</code>	master	Signal	Power management outputs of cluster 5.
<code>wake_request_6</code>	master	Signal	Power management outputs of cluster 6.
<code>wake_request_7</code>	master	Signal	Power management outputs of cluster 7.
<code>wake_request_8</code>	master	Signal	Power management outputs of cluster 8.
<code>wake_request_9</code>	master	Signal	Power management outputs of cluster 9.

## Parameters for GIC500\_Filter

### **GICD\_ITARGETSR-RAZWI**

If true, the GICD\_ITARGETS registers are **RAZ/WI**.

Type: `bool`

Default value: `false`

**ITS-count**

Number of Interrupt Translation Services to be instantiated (0=none).

Type: `uint8_t`

Default value: 1

**ITS-device-bits**

Number of bits supported for ITS device IDs.

Type: `int`

Default value: 16

**ITS-threaded-command-queue**

Enable execution of ITS commands in a separate thread which is sometimes required for cosimulation.

Type: `bool`

Default value: `true`

**SPI-count**

Number of SPIs that are implemented.

Type: `uint16_t`

Default value: 224

**clear-ISPENDR-bit-for-level-sensitive-interrupt-when-acknowledged**

If true, acknowledgement of level sensitive interrupt clears the ISPENDR register.

Type: `bool`

Default value: `false`

**cpus\_per\_cluster\_0**

Number of cores within cluster 0.

Type: `uint8_t`

Default value: 1

**cpus\_per\_cluster\_1**

Number of cores within cluster 1.

Type: `uint8_t`

Default value: 1

**cpus\_per\_cluster\_10**

Number of cores within cluster 10.

Type: `uint8_t`

Default value: 1

**cpus\_per\_cluster\_11**

Number of cores within cluster 11.

Type: `uint8_t`

Default value: 1

**cpus\_per\_cluster\_12**

Number of cores within cluster 12.

Type: `uint8_t`

Default value: 1

**cpus\_per\_cluster\_13**

Number of cores within cluster 13.

Type: `uint8_t`

Default value: 1

**cpus\_per\_cluster\_14**

Number of cores within cluster 14.

Type: `uint8_t`

Default value: 1

**cpus\_per\_cluster\_15**

Number of cores within cluster 15.

Type: `uint8_t`

Default value: 1

**cpus\_per\_cluster\_16**

Number of cores within cluster 16.

Type: `uint8_t`

Default value: 1

**cpus\_per\_cluster\_17**

Number of cores within cluster 17.

Type: uint8\_t

Default value: 1

**cpus\_per\_cluster\_18**

Number of cores within cluster 18.

Type: uint8\_t

Default value: 1

**cpus\_per\_cluster\_19**

Number of cores within cluster 19.

Type: uint8\_t

Default value: 1

**cpus\_per\_cluster\_2**

Number of cores within cluster 2.

Type: uint8\_t

Default value: 1

**cpus\_per\_cluster\_20**

Number of cores within cluster 20.

Type: uint8\_t

Default value: 1

**cpus\_per\_cluster\_21**

Number of cores within cluster 21.

Type: uint8\_t

Default value: 1

**cpus\_per\_cluster\_22**

Number of cores within cluster 22.

Type: uint8\_t

Default value: 1

**cpus\_per\_cluster\_23**

Number of cores within cluster 23.

Type: `uint8_t`

Default value: 1

**cpus\_per\_cluster\_24**

Number of cores within cluster 24.

Type: `uint8_t`

Default value: 1

**cpus\_per\_cluster\_25**

Number of cores within cluster 25.

Type: `uint8_t`

Default value: 1

**cpus\_per\_cluster\_26**

Number of cores within cluster 26.

Type: `uint8_t`

Default value: 1

**cpus\_per\_cluster\_27**

Number of cores within cluster 27.

Type: `uint8_t`

Default value: 1

**cpus\_per\_cluster\_28**

Number of cores within cluster 28.

Type: `uint8_t`

Default value: 1

**cpus\_per\_cluster\_29**

Number of cores within cluster 29.

Type: `uint8_t`



Default value: 1

**cpus\_per\_cluster\_3**

Number of cores within cluster 3.

Type: `uint8_t`

Default value: 1

**cpus\_per\_cluster\_30**

Number of cores within cluster 30.

Type: `uint8_t`

Default value: 1

**cpus\_per\_cluster\_31**

Number of cores within cluster 31.

Type: `uint8_t`

Default value: 1

**cpus\_per\_cluster\_4**

Number of cores within cluster 4.

Type: `uint8_t`

Default value: 1

**cpus\_per\_cluster\_5**

Number of cores within cluster 5.

Type: `uint8_t`

Default value: 1

**cpus\_per\_cluster\_6**

Number of cores within cluster 6.

Type: `uint8_t`

Default value: 1

**cpus\_per\_cluster\_7**

Number of cores within cluster 7.

Type: `uint8_t`

Default value: 1

**cpus\_per\_cluster\_8**

Number of cores within cluster 8.

Type: `uint8_t`

Default value: 1

**cpus\_per\_cluster\_9**

Number of cores within cluster 9.

Type: `uint8_t`

Default value: 1

**delay-ITS-accesses**

Delay accesses from the ITS until GICR\_SYNCRR is read.

Type: `bool`

Default value: `true`

**delay-redistributor-accesses**

Delay memory accesses from the redistributor until GICR\_SYNCRR is read.

Type: `bool`

Default value: `true`

**enable\_protocol\_checking**

Enable/disable protocol checking at cpu interface.

Type: `bool`

Default value: `false`

**enabled**

Enable GICv3 functionality; when false the component is inactive.

Type: `bool`

Default value: `true`

**has-two-security-states**

If true, has two security states.

Type: `bool`

Default value: `true`

### **num\_clusters**

Number of implemented affinity level1 clusters.

Type: `uint8_t`

Default value: `1`

### **print-memory-map**

Print memory map to stdout.

Type: `bool`

Default value: `false`

### **redistributor-threaded-sync**

Enable execution of redistributor delayed transactions in a separate thread which is sometimes required for cosimulation.

Type: `bool`

Default value: `true`

### **reg-base**

GIC500 base address.

Type: `uint64_t`

Default value: `0x2c010000`

### **using-generated-memorymap**

Use the generated memorymap for the GIC500 and warn if superfluous parameters are passed.

Type: `bool`

Default value: `true`

### **wakeup-on-reset**

Go against specification and start redistributors in woken-up state at reset. This allows software that was written for previous versions of the GICv3 specification to work correctly. This should not be used for production code or when the distributor is used separately from the core fast model.

Type: `bool`

Default value: `false`

## 3.188 GIC600

Defined in `LISA/GIC600.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r1p6	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### About GIC600

GIC600 and GIC600\_Filter are minimal models of an Arm GIC-600 Generic Interrupt Controller, suitable for single-chip or multichip systems, where multichip operation has been tested, with the following exceptions:

- LPI/LPI command is implemented but untested.
- Power operation is unimplemented.

They provide a simple configuration interface that allows designers to introduce GIC600-like functionality to their systems, while only implementing the architectural behavior, as defined by the GICv3 architecture.

All implementation-specific registers and functionality are implemented.

As with the other GIC components, there are two variants of the model with slightly different memory interfaces. Both GIC600 and GIC600\_Filter have a `pvbus_s` port for register accesses and a `pvbus_m` port for the LPI-related traffic from redistributors and the ITS. The `pvbus_m` port is also used to compose multichip operation.

In addition, the GIC600\_Filter variant has a `pvbus_filtermiss_m` port, to which any transactions on the `pvbus_s` port and not directed to a 4K page used by the GIC are forwarded unmodified. Such transactions are terminated in the component when using the GIC600 variant.

It is recommended to use the GIC600 variant in most cases.

Also, the RAS register is implemented and various RAS errors are reported through status registers. Not all the RAS errors are available because Fast Models does not model errors that can cause ECC errors. Fast Models supports error record 0 and error record 13+.



- Set the `FASTSIM_GIC_MEMORY_MAP` environment variable to 1 to print to stderr the memory map of any GICv3 or later models that are included in the platform being run.
- For writes to `GITS_TRANSLATE64R`, the MSIRewriter component can be used to create the correct data transactions based on the device ID. For more details about `GITS_TRANSLATE64R`, see [MSIRewriter](#).

## ManagerID, ExtendedID, and UserFlags

This table shows how this model encodes information in the `ManagerID`, `ExtendedID`, and `UserFlags` bus attributes:

**Table 3-649: ManagerID, ExtendedID, and UserFlags**

PVBus attribute	Bits used	Property encoded	Notes
ManagerID	63:0	Manager ID that invoked the register access	–
ExtendedID	–	–	Not used
UserFlags	–	–	Not used



Note

The `pvbus_m` port does not use `ManagerID`, `ExtendedID`, or `UserFlags`.

## Iris and MTI instances for GIC600

This model has the following Iris instances:

Name	Instance type
GIC600	GIC600
GIC600.GICV3_ProtocolChecker	GICv3ProtocolChecker
GIC600.ITS0	GICv3InterruptTranslationService
GIC600.ITS0.ExportTest.GIC600.ITS0.pvbus_m[0].pvbusmaster	PVBusMaster
GIC600.rd_0	GICv3RedistributorInternal
GIC600.rd_0_0	GICv3RedistributorInternal
GIC600.rd_0_0_0	GICv3RedistributorInternal
GIC600.rd_0_0_0_0	GICv3Redistributor
GIC600.rd_0_0_0_0.ExportTest.GIC600.rd_0_0_0_0.pvbus_m[0].pvbusmaster	PVBusMaster
GIC600.rd_t1	GICv3Distributor
GIC600.rd_t1.ExportTest.GIC600.rd_t1.pvbus_m[0].pvbusmaster	PVBusMaster

This model has the following MTI trace components:

Name	Component type
GIC600.GICV3_ProtocolChecker	GICv3ProtocolChecker
GIC600.ITS0	GICv3InterruptTranslationService
GIC600.ITS0.ExportTest.GIC600.ITS0.pvbus_m[0].pvbusmaster	PVBusMaster
GIC600.rd_0	GICv3RedistributorInternal
GIC600.rd_0_0	GICv3RedistributorInternal
GIC600.rd_0_0_0	GICv3RedistributorInternal
GIC600.rd_0_0_0_0	GICv3Redistributor
GIC600.rd_0_0_0_0.ExportTest.GIC600.rd_0_0_0_0.pvbus_m[0].pvbusmaster	PVBusMaster
GIC600.rd_t1	GICv3Distributor

Name	Component type
GIC600.rd_tl.ExportTest.GIC600.rd_tl.pvbus_m[0].pvbusmaster	PVBusMaster

## Ports for GIC600

Port	Direction	Protocol	Description
chip_id	slave	Value	chip_id port which is valid from GIC600 r1p2. Writing to this port for prior GIC600 version will be ignored.
cpu_active_s	slave	Signal	CPUActive pins.
po_reset	slave	Signal	Resets.
ppi_in_0	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 0.
ppi_in_100	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 100.
ppi_in_101	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 101.
ppi_in_102	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 102.
ppi_in_103	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 103.
ppi_in_104	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 104.
ppi_in_105	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 105.
ppi_in_106	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 106.
ppi_in_107	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 107.
ppi_in_108	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 108.
ppi_in_109	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 109.
ppi_in_10	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 10.
ppi_in_110	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 110.
ppi_in_111	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 111.
ppi_in_112	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 112.
ppi_in_113	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 113.
ppi_in_114	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 114.
ppi_in_115	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 115.
ppi_in_116	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 116.
ppi_in_117	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 117.
ppi_in_118	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 118.
ppi_in_119	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 119.
ppi_in_11	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 11.
ppi_in_120	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 120.
ppi_in_121	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 121.
ppi_in_122	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 122.
ppi_in_123	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 123.
ppi_in_124	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 124.
ppi_in_125	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 125.
ppi_in_126	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 126.
ppi_in_127	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 127.
ppi_in_128	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 128.

Port	Direction	Protocol	Description
ppi_in_129	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 129.
ppi_in_12	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 12.
ppi_in_130	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 130.
ppi_in_131	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 131.
ppi_in_132	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 132.
ppi_in_133	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 133.
ppi_in_134	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 134.
ppi_in_135	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 135.
ppi_in_136	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 136.
ppi_in_137	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 137.
ppi_in_138	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 138.
ppi_in_139	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 139.
ppi_in_13	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 13.
ppi_in_140	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 140.
ppi_in_141	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 141.
ppi_in_142	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 142.
ppi_in_143	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 143.
ppi_in_144	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 144.
ppi_in_145	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 145.
ppi_in_146	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 146.
ppi_in_147	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 147.
ppi_in_148	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 148.
ppi_in_149	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 149.
ppi_in_14	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 14.
ppi_in_150	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 150.
ppi_in_151	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 151.
ppi_in_152	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 152.
ppi_in_153	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 153.
ppi_in_154	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 154.
ppi_in_155	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 155.
ppi_in_156	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 156.
ppi_in_157	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 157.
ppi_in_158	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 158.
ppi_in_159	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 159.
ppi_in_15	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 15.
ppi_in_160	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 160.
ppi_in_161	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 161.
ppi_in_162	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 162.
ppi_in_163	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 163.
ppi_in_164	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 164.

Port	Direction	Protocol	Description
ppi_in_165	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 165.
ppi_in_166	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 166.
ppi_in_167	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 167.
ppi_in_168	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 168.
ppi_in_169	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 169.
ppi_in_16	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 16.
ppi_in_170	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 170.
ppi_in_171	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 171.
ppi_in_172	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 172.
ppi_in_173	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 173.
ppi_in_174	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 174.
ppi_in_175	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 175.
ppi_in_176	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 176.
ppi_in_177	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 177.
ppi_in_178	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 178.
ppi_in_179	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 179.
ppi_in_17	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 17.
ppi_in_180	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 180.
ppi_in_181	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 181.
ppi_in_182	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 182.
ppi_in_183	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 183.
ppi_in_184	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 184.
ppi_in_185	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 185.
ppi_in_186	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 186.
ppi_in_187	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 187.
ppi_in_188	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 188.
ppi_in_189	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 189.
ppi_in_18	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 18.
ppi_in_190	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 190.
ppi_in_191	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 191.
ppi_in_192	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 192.
ppi_in_193	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 193.
ppi_in_194	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 194.
ppi_in_195	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 195.
ppi_in_196	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 196.
ppi_in_197	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 197.
ppi_in_198	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 198.
ppi_in_199	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 199.
ppi_in_19	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 19.
ppi_in_1	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 1.



Port	Direction	Protocol	Description
ppi_in_200	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 200.
ppi_in_201	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 201.
ppi_in_202	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 202.
ppi_in_203	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 203.
ppi_in_204	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 204.
ppi_in_205	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 205.
ppi_in_206	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 206.
ppi_in_207	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 207.
ppi_in_208	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 208.
ppi_in_209	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 209.
ppi_in_20	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 20.
ppi_in_210	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 210.
ppi_in_211	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 211.
ppi_in_212	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 212.
ppi_in_213	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 213.
ppi_in_214	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 214.
ppi_in_215	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 215.
ppi_in_216	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 216.
ppi_in_217	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 217.
ppi_in_218	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 218.
ppi_in_219	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 219.
ppi_in_21	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 21.
ppi_in_220	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 220.
ppi_in_221	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 221.
ppi_in_222	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 222.
ppi_in_223	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 223.
ppi_in_224	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 224.
ppi_in_225	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 225.
ppi_in_226	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 226.
ppi_in_227	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 227.
ppi_in_228	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 228.
ppi_in_229	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 229.
ppi_in_22	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 22.
ppi_in_230	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 230.
ppi_in_231	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 231.
ppi_in_232	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 232.
ppi_in_233	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 233.
ppi_in_234	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 234.
ppi_in_235	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 235.
ppi_in_236	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 236.

Port	Direction	Protocol	Description
ppi_in_237	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 237.
ppi_in_238	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 238.
ppi_in_239	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 239.
ppi_in_23	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 23.
ppi_in_240	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 240.
ppi_in_241	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 241.
ppi_in_242	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 242.
ppi_in_243	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 243.
ppi_in_244	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 244.
ppi_in_245	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 245.
ppi_in_246	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 246.
ppi_in_247	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 247.
ppi_in_248	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 248.
ppi_in_249	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 249.
ppi_in_24	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 24.
ppi_in_250	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 250.
ppi_in_251	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 251.
ppi_in_252	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 252.
ppi_in_253	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 253.
ppi_in_254	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 254.
ppi_in_255	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 255.
ppi_in_25	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 25.
ppi_in_26	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 26.
ppi_in_27	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 27.
ppi_in_28	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 28.
ppi_in_29	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 29.
ppi_in_2	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 2.
ppi_in_30	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 30.
ppi_in_31	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 31.
ppi_in_32	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 32.
ppi_in_33	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 33.
ppi_in_34	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 34.
ppi_in_35	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 35.
ppi_in_36	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 36.
ppi_in_37	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 37.
ppi_in_38	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 38.
ppi_in_39	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 39.
ppi_in_3	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 3.
ppi_in_40	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 40.
ppi_in_41	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 41.

Port	Direction	Protocol	Description
ppi_in_42	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 42.
ppi_in_43	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 43.
ppi_in_44	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 44.
ppi_in_45	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 45.
ppi_in_46	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 46.
ppi_in_47	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 47.
ppi_in_48	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 48.
ppi_in_49	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 49.
ppi_in_4	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 4.
ppi_in_50	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 50.
ppi_in_51	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 51.
ppi_in_52	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 52.
ppi_in_53	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 53.
ppi_in_54	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 54.
ppi_in_55	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 55.
ppi_in_56	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 56.
ppi_in_57	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 57.
ppi_in_58	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 58.
ppi_in_59	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 59.
ppi_in_5	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 5.
ppi_in_60	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 60.
ppi_in_61	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 61.
ppi_in_62	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 62.
ppi_in_63	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 63.
ppi_in_64	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 64.
ppi_in_65	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 65.
ppi_in_66	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 66.
ppi_in_67	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 67.
ppi_in_68	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 68.
ppi_in_69	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 69.
ppi_in_6	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 6.
ppi_in_70	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 70.
ppi_in_71	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 71.
ppi_in_72	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 72.
ppi_in_73	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 73.
ppi_in_74	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 74.
ppi_in_75	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 75.
ppi_in_76	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 76.
ppi_in_77	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 77.
ppi_in_78	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 78.

Port	Direction	Protocol	Description
ppi_in_79	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 79.
ppi_in_7	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 7.
ppi_in_80	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 80.
ppi_in_81	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 81.
ppi_in_82	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 82.
ppi_in_83	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 83.
ppi_in_84	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 84.
ppi_in_85	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 85.
ppi_in_86	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 86.
ppi_in_87	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 87.
ppi_in_88	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 88.
ppi_in_89	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 89.
ppi_in_8	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 8.
ppi_in_90	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 90.
ppi_in_91	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 91.
ppi_in_92	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 92.
ppi_in_93	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 93.
ppi_in_94	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 94.
ppi_in_95	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 95.
ppi_in_96	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 96.
ppi_in_97	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 97.
ppi_in_98	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 98.
ppi_in_99	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 99.
ppi_in_9	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 9.
pvbuss_m	master	PVBus	Memory bus out: transactions generated by the IRI.
pvbuss_s	slave	PVBus	Memory bus in: memory-mapped register accesses are accepted on this interface.
redistributor_m	master	GICv3Comms	Input from and output to CPU interface.
reset	slave	Signal	Resets.
spi_in	slave	Signal	Shared peripheral interrupts.
wake_request	master	Signal	Power management outputs.

## Parameters for GIC600

### CPU-affinities

A comma separated list of dotted quads containing the affinities of all PEs connected to this IRI. If CPU-affinities-file is specified, this parameter is ignored.

Type: string

Default value: N/A

**CPU-affinities-file**

A file containing comma separated list of dotted quads containing the affinities of all PEs connected to this IRL. If this parameter is specified, CPU-affinities parameter will be ignored even when it is given.

Type: `string`

Default value: `N/A`

**DS-behaviour**

GICD\_CTLR.DS field behaviour

0 :**RAZ/WI**

1 :**RAO/WI**

2 :**RW**.

Type: `int`

Default value: 2

**IIDR**

GICD\_IIDR, GICR\_IIDR and GITS\_IIDR value. Defaults to the latest revision.

Type: `uint32_t`

Default value: `0x0201743B`

**ITS-ID-bits**

Number of interrupt bits supported by ITS.

Type: `uint8_t`

Default value: 16

**ITS-collection-ID-bits**

Number of collection bits supported by ITS (optional parameter, 0 => 16bits support and GITS\_TYPER.CIL=0).

Type: `uint8_t`

Default value: 8

**ITS-count**

Number of Interrupt Translation Services to be instantiated (0=none).

Type: `uint8_t`

Default value: 1

**ITS-device-bits**

Number of bits supported for ITS device IDs.

Type: `uint8_t`

Default value: 16

**PPI-count**

Selects the number of PPI available for each PE:

**8**

id22-27,29,30

**12**

id 20-31

**16**

id 16-31.

Type: `int`

Default value: 16

**RAS-CFI-support**

If true, fault handling interrupt for corrected errors is supported. Not supported otherwise.

Type: `bool`

Default value: `false`

**RAS-FI-support**

If true, fault handling interrupt is supported. Not supported otherwise.

Type: `bool`

Default value: `false`

**RAS-UE-support**

If true, In-band uncorrected error reporting is supported. Not supported otherwise.

Type: `bool`

Default value: `false`

**RAS-UI-support**

If true, error recovery interrupt for uncorrected errors is supported. Not supported otherwise.

Type: `bool`

Default value: `false`

### **SPI-blocks**

Number of SPI blocks supported by the IRI, each block contains 32 SPIs.

Type: `int`

Default value: 30

### **affinity-width**

A dotted quad indicating the bitwidth of fields at each affinity level.

Type: `string`

Default value: `"4.8.8.8"`

### **chip-id**

Chip ID when multichip operation is enabled.

Type: `uint8_t`

Default value: 0

### **chip-select-affinity-level**

Affinity level 2 or 3 can be used for chip select.

Type: `int`

Default value: 3

### **clear-ISPENDR-bit-for-level-sensitive-interrupt-when-acknowledged**

If true, acknowledgement of a level sensitive interrupt clears the corresponding bit in the ISPENDR register.

Type: `bool`

Default value: `false`

### **direct-lpi-support**

Enable support for LPI operations through GICR registers.

Type: `bool`

Default value: `false`

**enable-multichip-operation**

Enables multi-chip operation between Distributors in distributed GIC IRI.

Type: `bool`

Default value: `true`

**enabled**

Enable GICv3 functionality; when false the component is inactive.

Type: `bool`

Default value: `true`

**gicp-allow-ns-reset**

If true, non-secure read/write access to GICP register is allowed at reset. Not allowed otherwise. This emulates `gicp_allow_ns` tie-off signal.

Type: `bool`

Default value: `true`

**gict-allow-ns-reset**

If true, non-secure read/write access to GICT register is allowed at reset. Not allowed otherwise. This emulates `gict_allow_ns` tie-off signal.

Type: `bool`

Default value: `true`

**max-cores-supported-by-GCI**

GCI can support 1-64 cores. Maximum supported cores in GCI is configurable by device.

Type: `int`

Default value: 64

**max-pe-on-chip**

Maximum number of cores on any single chip. This will be used to identify the target chip and core when multichip operation is enabled.

Type: `int`

Default value: 4



**multichip-threaded-dgi**

Enable sending of multichip DGI messages in a separate thread, when multichip operation is enabled.

Type: `bool`

Default value: `true`

**print-memory-map**

Print memory map to stdout.

Type: `bool`

Default value: `false`

**redistributor-group**

Redistributor grouping information with affinity as JSON :

```
{
  "0": [
    "0.0.0.0",
    "0.0.0.1"
  ],
  "1": [
    "0.0.1.0",
    "0.0.1.1"
  ]
}
```

where RD with quad 0.0.0.0 and 0.0.0.1 belongs to RD group 0. All the RDs belong to one group when this parameter is not given.

Type: `string`

Default value: `N/A`

**redistributor-group-file**

File path to redistributor grouping information with affinity as JSON.

The file uses the same format as `redistributor-group` parameter.

Type: `string`

Default value: `N/A`

**redistributor-power-managed-by-pwrr**

GIC600 redistributor power management is done by updating GICR\_PWRR register.

Type: `bool`

Default value: `true`

**reg-base**

GIC-600 base address.

Type: `uint64_t`

Default value: `0x2c010000`

**reg-base-per-redistributor**

Base address for each redistributor in the form:

```
0.0.0.0=0x2c010000, 0.0.0.1=0x2c020000
```

All redistributors must be specified and this overrides the `reg-base` parameter (except that `reg-base` will still be used for the top-level redistributor). If `reg-base-per-redistributor-file` is specified, this parameter is ignored.

Type: `string`

Default value: `""`

## 3.189 GIC600AE

Defined in `LISA/GIC600AE.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p2	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### About GIC600AE

GIC600AE and GIC600AE\_Filter are minimal models of an Arm GIC-600AE Generic Interrupt Controller, suitable for single-chip or multichip systems, where multichip operation has been tested, with the following exceptions:

- LPI/LPI command is implemented but untested.
- Power operation is unimplemented.

They provide a simple configuration interface that allows designers to introduce GIC600AE-like functionality to their systems, while only implementing the architectural behavior, as defined by the GICv3 architecture.

All implementation-specific registers and functionality are implemented.

As with the other GIC components, there are two variants of the model with slightly different memory interfaces. Both GIC600AE and GIC600AE\_Filter have a `pvbus_s` port for register accesses and a `pvbus_m` port for the LPI-related traffic from redistributors and the ITS. The `pvbus_m` port is also used to compose multichip operation.

In addition, the GIC600AE\_Filter variant has a `pvbus_filtermiss_m` port, to which any transactions on the `pvbus_s` port and not directed to a 4 KB page used by the GIC are forwarded unmodified. Such transactions are terminated in the component when using the GIC600AE variant.

It is recommended to use the GIC600AE variant in most cases.

Also, the RAS register is implemented and various RAS errors are reported through status registers. Not all the RAS errors are available because Fast Models does not model errors that can cause ECC errors. Fast Models supports error record 0 and error record 13+. FMU registers are also implemented and are accessible by software.



Note

- Set the `FASTSIM_GIC_MEMORY_MAP` environment variable to 1 to print to stderr the memory map of any GICv3 or later models that are included in the platform being run.
- For writes to GITS\_TRANSLATE64R, the MSIRewriter component can be used to create the correct data transactions based on the device ID. For more details about GITS\_TRANSLATE64R, see [MSIRewriter](#).

## ManagerID, ExtendedID, and UserFlags

This table shows how this model encodes information in the `ManagerID`, `ExtendedID`, and `UserFlags` bus attributes:

**Table 3-654: ManagerID, ExtendedID, and UserFlags**

PVBus attribute	Bits used	Property encoded	Notes
ManagerID	63:0	Manager ID that invoked the register access	–
ExtendedID	–	–	Not used
UserFlags	–	–	Not used



Note

The `pvbus_m` port does not use `ManagerID`, `ExtendedID`, or `UserFlags`.

## AE-specific features implemented

GIC600AE is a Functional Safety (FuSa) variant of GIC600. It has the following differences from GIC600:

- Both GIC600AE and GIC600 support RAS, but only GIC600AE supports the Fault Management Unit (FMU). This support is controlled by the `has-fmu` parameter which is true by default. The FMU resides in the Distributor and processes faults that are detected by the Safety Mechanisms from all blocks.

- In GIC600AE, the GIC reset pin is connected to FMU reset. Only reset changes the error record registers and FMU\_ERRGSR to their reset values.
- In GIC600AE, the Safety Mechanism detects faults and forwards them to the FMU. The FMU forwards all errors to the Safety Island.
- The APB port has been added to GIC600AE for FuSa purposes. It does not exist on the GIC600.
- GIC600AE supports both Error Recovery Interrupt (ERI) and Fault Handling Interrupt (FHI).
- GIC600AE has the limitation that it only supports error injection through the FMU\_SMINJERR register.

### Iris and MTI instances for GIC600AE

This model has the following Iris instances:

Name	Instance type
GIC600AE	GIC600AE
GIC600AE.GICV3_ProtocolChecker	GICv3ProtocolChecker
GIC600AE.ITS0	GICv3InterruptTranslationService
GIC600AE.ITS0.ExportTest.GIC600AE.ITS0.pvbus_m[0].pvbusmaster	PVBusMaster
GIC600AE.rd_0	GICv3RedistributorInternal
GIC600AE.rd_0_0	GICv3RedistributorInternal
GIC600AE.rd_0_0_0	GICv3RedistributorInternal
GIC600AE.rd_0_0_0_0	GICv3Redistributor
GIC600AE.rd_0_0_0_0.ExportTest.GIC600AE.rd_0_0_0_0.pvbus_m[0].pvbusmaster	PVBusMaster
GIC600AE.rd_t1	GICv3Distributor
GIC600AE.rd_t1.ExportTest.GIC600AE.rd_t1.pvbus_m[0].pvbusmaster	PVBusMaster

This model has the following MTI trace components:

Name	Component type
GIC600AE.GICV3_ProtocolChecker	GICv3ProtocolChecker
GIC600AE.ITS0	GICv3InterruptTranslationService
GIC600AE.ITS0.ExportTest.GIC600AE.ITS0.pvbus_m[0].pvbusmaster	PVBusMaster
GIC600AE.rd_0	GICv3RedistributorInternal
GIC600AE.rd_0_0	GICv3RedistributorInternal
GIC600AE.rd_0_0_0	GICv3RedistributorInternal
GIC600AE.rd_0_0_0_0	GICv3Redistributor
GIC600AE.rd_0_0_0_0.ExportTest.GIC600AE.rd_0_0_0_0.pvbus_m[0].pvbusmaster	PVBusMaster
GIC600AE.rd_t1	GICv3Distributor
GIC600AE.rd_t1.ExportTest.GIC600AE.rd_t1.pvbus_m[0].pvbusmaster	PVBusMaster

## Ports for GIC600AE

Port	Direction	Protocol	Description
apb_bus	slave	PVBus	APB4 port to FMU, which will be connected from safety island for FuSa
chip_id	slave	Value	chip_id port used for multichip operation
cpu_active_s	slave	Signal	CPUActive pins.
err_int	master	Signal	RAS Error Recovery Interrupt
fault_int	master	Signal	RAS Fault Handling Interrupt
fm_u_error_int	master	Signal	FuSa FMU error interrupt signal
fm_u_fault_int	master	Signal	FuSa FMU fault interrupt signal
po_reset	slave	Signal	Resets. This is used as a dbg_reset in TRM, and also be used for cold reset for PMU and FMU.
ppi_in_0	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 0.
ppi_in_100	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 100.
ppi_in_101	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 101.
ppi_in_102	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 102.
ppi_in_103	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 103.
ppi_in_104	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 104.
ppi_in_105	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 105.
ppi_in_106	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 106.
ppi_in_107	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 107.
ppi_in_108	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 108.
ppi_in_109	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 109.
ppi_in_110	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 110.
ppi_in_111	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 111.
ppi_in_112	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 112.
ppi_in_113	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 113.
ppi_in_114	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 114.
ppi_in_115	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 115.
ppi_in_116	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 116.
ppi_in_117	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 117.
ppi_in_118	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 118.
ppi_in_119	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 119.
ppi_in_120	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 120.
ppi_in_121	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 121.
ppi_in_122	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 122.
ppi_in_123	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 123.
ppi_in_124	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 124.
ppi_in_125	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 125.
ppi_in_126	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 126.

Port	Direction	Protocol	Description
ppi_in_127	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 127.
ppi_in_128	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 128.
ppi_in_129	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 129.
ppi_in_12	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 12.
ppi_in_130	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 130.
ppi_in_131	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 131.
ppi_in_132	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 132.
ppi_in_133	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 133.
ppi_in_134	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 134.
ppi_in_135	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 135.
ppi_in_136	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 136.
ppi_in_137	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 137.
ppi_in_138	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 138.
ppi_in_139	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 139.
ppi_in_13	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 13.
ppi_in_140	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 140.
ppi_in_141	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 141.
ppi_in_142	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 142.
ppi_in_143	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 143.
ppi_in_144	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 144.
ppi_in_145	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 145.
ppi_in_146	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 146.
ppi_in_147	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 147.
ppi_in_148	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 148.
ppi_in_149	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 149.
ppi_in_14	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 14.
ppi_in_150	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 150.
ppi_in_151	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 151.
ppi_in_152	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 152.
ppi_in_153	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 153.
ppi_in_154	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 154.
ppi_in_155	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 155.
ppi_in_156	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 156.
ppi_in_157	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 157.
ppi_in_158	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 158.
ppi_in_159	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 159.
ppi_in_15	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 15.
ppi_in_160	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 160.
ppi_in_161	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 161.
ppi_in_162	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 162.

Port	Direction	Protocol	Description
ppi_in_163	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 163.
ppi_in_164	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 164.
ppi_in_165	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 165.
ppi_in_166	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 166.
ppi_in_167	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 167.
ppi_in_168	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 168.
ppi_in_169	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 169.
ppi_in_16	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 16.
ppi_in_170	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 170.
ppi_in_171	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 171.
ppi_in_172	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 172.
ppi_in_173	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 173.
ppi_in_174	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 174.
ppi_in_175	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 175.
ppi_in_176	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 176.
ppi_in_177	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 177.
ppi_in_178	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 178.
ppi_in_179	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 179.
ppi_in_17	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 17.
ppi_in_180	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 180.
ppi_in_181	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 181.
ppi_in_182	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 182.
ppi_in_183	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 183.
ppi_in_184	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 184.
ppi_in_185	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 185.
ppi_in_186	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 186.
ppi_in_187	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 187.
ppi_in_188	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 188.
ppi_in_189	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 189.
ppi_in_18	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 18.
ppi_in_190	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 190.
ppi_in_191	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 191.
ppi_in_192	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 192.
ppi_in_193	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 193.
ppi_in_194	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 194.
ppi_in_195	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 195.
ppi_in_196	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 196.
ppi_in_197	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 197.
ppi_in_198	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 198.
ppi_in_199	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 199.

Port	Direction	Protocol	Description
ppi_in_19	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 19.
ppi_in_1	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 1.
ppi_in_200	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 200.
ppi_in_201	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 201.
ppi_in_202	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 202.
ppi_in_203	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 203.
ppi_in_204	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 204.
ppi_in_205	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 205.
ppi_in_206	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 206.
ppi_in_207	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 207.
ppi_in_208	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 208.
ppi_in_209	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 209.
ppi_in_20	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 20.
ppi_in_210	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 210.
ppi_in_211	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 211.
ppi_in_212	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 212.
ppi_in_213	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 213.
ppi_in_214	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 214.
ppi_in_215	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 215.
ppi_in_216	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 216.
ppi_in_217	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 217.
ppi_in_218	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 218.
ppi_in_219	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 219.
ppi_in_21	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 21.
ppi_in_220	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 220.
ppi_in_221	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 221.
ppi_in_222	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 222.
ppi_in_223	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 223.
ppi_in_224	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 224.
ppi_in_225	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 225.
ppi_in_226	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 226.
ppi_in_227	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 227.
ppi_in_228	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 228.
ppi_in_229	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 229.
ppi_in_22	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 22.
ppi_in_230	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 230.
ppi_in_231	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 231.
ppi_in_232	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 232.
ppi_in_233	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 233.
ppi_in_234	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 234.



Port	Direction	Protocol	Description
ppi_in_235	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 235.
ppi_in_236	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 236.
ppi_in_237	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 237.
ppi_in_238	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 238.
ppi_in_239	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 239.
ppi_in_23	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 23.
ppi_in_240	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 240.
ppi_in_241	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 241.
ppi_in_242	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 242.
ppi_in_243	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 243.
ppi_in_244	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 244.
ppi_in_245	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 245.
ppi_in_246	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 246.
ppi_in_247	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 247.
ppi_in_248	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 248.
ppi_in_249	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 249.
ppi_in_24	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 24.
ppi_in_250	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 250.
ppi_in_251	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 251.
ppi_in_252	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 252.
ppi_in_253	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 253.
ppi_in_254	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 254.
ppi_in_255	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 255.
ppi_in_25	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 25.
ppi_in_26	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 26.
ppi_in_27	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 27.
ppi_in_28	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 28.
ppi_in_29	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 29.
ppi_in_2	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 2.
ppi_in_30	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 30.
ppi_in_31	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 31.
ppi_in_32	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 32.
ppi_in_33	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 33.
ppi_in_34	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 34.
ppi_in_35	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 35.
ppi_in_36	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 36.
ppi_in_37	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 37.
ppi_in_38	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 38.
ppi_in_39	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 39.
ppi_in_3	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 3.

Port	Direction	Protocol	Description
ppi_in_40	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 40.
ppi_in_41	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 41.
ppi_in_42	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 42.
ppi_in_43	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 43.
ppi_in_44	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 44.
ppi_in_45	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 45.
ppi_in_46	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 46.
ppi_in_47	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 47.
ppi_in_48	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 48.
ppi_in_49	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 49.
ppi_in_4	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 4.
ppi_in_50	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 50.
ppi_in_51	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 51.
ppi_in_52	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 52.
ppi_in_53	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 53.
ppi_in_54	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 54.
ppi_in_55	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 55.
ppi_in_56	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 56.
ppi_in_57	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 57.
ppi_in_58	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 58.
ppi_in_59	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 59.
ppi_in_5	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 5.
ppi_in_60	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 60.
ppi_in_61	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 61.
ppi_in_62	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 62.
ppi_in_63	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 63.
ppi_in_64	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 64.
ppi_in_65	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 65.
ppi_in_66	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 66.
ppi_in_67	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 67.
ppi_in_68	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 68.
ppi_in_69	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 69.
ppi_in_6	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 6.
ppi_in_70	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 70.
ppi_in_71	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 71.
ppi_in_72	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 72.
ppi_in_73	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 73.
ppi_in_74	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 74.
ppi_in_75	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 75.
ppi_in_76	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 76.

Port	Direction	Protocol	Description
ppi_in_77	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 77.
ppi_in_78	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 78.
ppi_in_79	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 79.
ppi_in_7	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 7.
ppi_in_80	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 80.
ppi_in_81	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 81.
ppi_in_82	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 82.
ppi_in_83	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 83.
ppi_in_84	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 84.
ppi_in_85	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 85.
ppi_in_86	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 86.
ppi_in_87	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 87.
ppi_in_88	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 88.
ppi_in_89	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 89.
ppi_in_8	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 8.
ppi_in_90	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 90.
ppi_in_91	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 91.
ppi_in_92	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 92.
ppi_in_93	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 93.
ppi_in_94	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 94.
ppi_in_95	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 95.
ppi_in_96	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 96.
ppi_in_97	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 97.
ppi_in_98	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 98.
ppi_in_99	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 99.
ppi_in_9	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 9.
pvbus_m	master	PVBus	Memory bus out: transactions generated by the IRI.
pvbus_s	slave	PVBus	Memory bus in: memory-mapped register accesses are accepted on this interface.
redistributor_m	master	GICv3Comms	Input from and output to CPU interface.
reset	slave	Signal	Resets.
spi_in	slave	Signal	Shared peripheral interrupts.
wake_request	master	Signal	Power management outputs.

## Parameters for GIC600AE

### CPU-affinities

A comma separated list of dotted quads containing the affinities of all PEs connected to this IRI. If CPU-affinities-file is specified, this parameter is ignored.

Type: string

Default value: N/A

**CPU-affinities-file**

A file containing comma separated list of dotted quads containing the affinities of all PEs connected to this IRL. If this parameter is specified, CPU-affinities parameter will be ignored even when it is given.

Type: `string`

Default value: `N/A`

**DS-behaviour**

GICD\_CTLR.DS field behaviour

0 :**RAZ/WI**

1 :**RAO/WI**

2 :**RW**.

Type: `int`

Default value: 2

**IIDR**

GICD\_IIDR, GICR\_IIDR and GITS\_IIDR value. Defaults to the latest revision.

Type: `uint32_t`

Default value: `0x0300543B`

**ITS-ID-bits**

Number of interrupt bits supported by ITS.

Type: `uint8_t`

Default value: 16

**ITS-collection-ID-bits**

Number of collection bits supported by ITS (optional parameter, 0 => 16bits support and GITS\_TYPER.CIL=0).

Type: `uint8_t`

Default value: 8

**ITS-count**

Number of Interrupt Translation Services to be instantiated (0=none).

Type: `uint8_t`

Default value: 1

**ITS-device-bits**

Number of bits supported for ITS device IDs.

Type: `uint8_t`

Default value: 16

**PPI-count**

Selects the number of PPI available for each PE

**8**

id22-27,29,30

**12**

id 20-31

**16**

id 16-31.

Type: `int`

Default value: 16

**SPI-blocks**

Number of SPI blocks supported by the IRI, each block contains 32 SPIs.

Type: `int`

Default value: 30

**affinity-width**

A dotted quad indicating the bitwidth of fields at each affinity level.

Type: `string`

Default value: "4.8.8.8"

**chip-id**

Chip ID for multichip operation.

Type: `uint8_t`

Default value: 0

**chip-select-affinity-level**

Affinity level 2 or 3 can be used for chip select.

Type: `int`

Default value: `3`

### **clear-ISPENDR-bit-for-level-sensitive-interrupt-when-acknowledged**

If true, acknowledgement of a level sensitive interrupt clears the corresponding bit in the ISPENDR register.

Type: `bool`

Default value: `false`

### **direct-lpi-support**

Enable support for LPI operations through GICR registers.

Type: `bool`

Default value: `false`

### **enable-multichip-operation**

Enables multi-chip operation between Distributors in distributed GIC IRI.

Type: `bool`

Default value: `true`

### **enabled**

Enable GICv3 functionality; when false the component is inactive.

Type: `bool`

Default value: `true`

### **gicp-allow-ns-reset**

If true, non-secure read/write access to GICP register is allowed at reset. Not allowed otherwise. This emulates gicp\_allow\_ns tie-off signal.

Type: `bool`

Default value: `true`

### **gict-allow-ns-reset**

If true, non-secure read/write access to GICT register is allowed at reset. Not allowed otherwise. This emulates gict\_allow\_ns tie-off signal.

Type: `bool`

Default value: `true`

**max-cores-supported-by-GCI**

GCI can support 1-64 cores. Maximum supported cores in GCI is configurable by device.

Type: `int`

Default value: 64

**max-pe-on-chip**

Maximum number of cores on any single chip. This will be used to identify the target chip and core for multichip operation.

Type: `int`

Default value: 4

**multichip-threaded-dgi**

Enable sending of multichip DGI messages in a separate thread.

Type: `bool`

Default value: `true`

**print-memory-map**

Print memory map to stdout.

Type: `bool`

Default value: `false`

**redistributor-group**

Redistributor grouping information with affinity as JSON :

```
{
  "0":
    ["0.0.0.0",
     "0.0.0.1"],
  "1":
    ["0.0.1.0",
     "0.0.1.1"]
}
```

where RD with quad 0.0.0.0 and 0.0.0.1 belongs to RD group 0. All the RDs belong to one group when this parameter is not given.

Type: `string`

Default value: `N/A`

**redistributor-group-file**

File path to redistributor grouping information with affinity as JSON.

The file uses the same format as `redistributor-group` parameter.

Type: `string`

Default value: `N/A`

**redistributor-power-managed-by-pwrr**

GIC600AE redistributor power management is done by updating GICR\_PWRR register.

Type: `bool`

Default value: `true`

**reg-base**

GIC600AE base address.

Type: `uint64_t`

Default value: `0x2c010000`

**reg-base-per-redistributor**

Base address for each redistributor in the form:

```
0.0.0.0=0x2c010000, 0.0.0.1=0x2c020000
```

All redistributors must be specified and this overrides the `reg-base` parameter (except that `reg-base` will still be used for the top-level redistributor). If `reg-base-per-redistributor-file` is specified, this parameter is ignored.

Type: `string`

Default value: `""`

3.190 GIC600AE\_Filter

Defined in `LISA/GIC600AE.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [Quality level definitions](#).



## About GIC600AE\_Filter

GIC600AE and GIC600AE\_Filter are minimal models of an Arm GIC-600AE Generic Interrupt Controller, suitable for single-chip or multichip systems, where multichip operation has been tested, with the following exceptions:

- LPI/LPI command is implemented but untested.
- Power operation is unimplemented.

They provide a simple configuration interface that allows designers to introduce GIC600AE-like functionality to their systems, while only implementing the architectural behavior, as defined by the GICv3 architecture.

All implementation-specific registers and functionality are implemented.

As with the other GIC components, there are two variants of the model with slightly different memory interfaces. Both GIC600AE and GIC600AE\_Filter have a `pvbuss_s` port for register accesses and a `pvbuss_m` port for the LPI-related traffic from redistributors and the ITS. The `pvbuss_m` port is also used to compose multichip operation.

In addition, the GIC600AE\_Filter variant has a `pvbuss_filtermiss_m` port, to which any transactions on the `pvbuss_s` port and not directed to a 4 KB page used by the GIC are forwarded unmodified. Such transactions are terminated in the component when using the GIC600AE variant.

It is recommended to use the GIC600AE variant in most cases.

Also, the RAS register is implemented and various RAS errors are reported through status registers. Not all the RAS errors are available because Fast Models does not model errors that can cause ECC errors. Fast Models supports error record 0 and error record 13+. FMU registers are also implemented and are accessible by software.



Note

- Set the FASTSIM\_GIC\_MEMORY\_MAP environment variable to 1 to print to stderr the memory map of any GICv3 or later models that are included in the platform being run.
- For writes to `GITS_TRANSLATE64R`, the MSIRewriter component can be used to create the correct data transactions based on the device ID. For more details about `GITS_TRANSLATE64R`, see [MSIRewriter](#).

## ManagerID, ExtendedID, and UserFlags

This table shows how this model encodes information in the `ManagerID`, `ExtendedID`, and `UserFlags` bus attributes:

**Table 3-659: ManagerID, ExtendedID, and UserFlags**

PVBus attribute	Bits used	Property encoded	Notes
ManagerID	63:0	Manager ID that invoked the register access	–
ExtendedID	–	–	Not used
UserFlags	–	–	Not used



The `pvbus_m` port does not use `ManagerID`, `ExtendedID`, Or `UserFlags`.

## AE-specific features implemented

GIC600AE\_Filter is a Functional Safety (FuSa) variant of GIC600\_Filter. It has the following differences from GIC600\_Filter:

- Both GIC600AE\_Filter and GIC600\_Filter support RAS, but only GIC600AE\_Filter supports the Fault Management Unit (FMU). This support is controlled by the `has-fmu` parameter which is true by default. The FMU resides in the Distributor and processes faults that are detected by the Safety Mechanisms from all blocks.
- In GIC600AE\_Filter, the GIC reset pin is connected to FMU reset. Only reset changes the error record registers and FMU\_ERRGSR to their reset values.
- In GIC600AE\_Filter, the Safety Mechanism detects faults and forwards them to the FMU. The FMU forwards all errors to the Safety Island.
- The APB port has been added to GIC600AE\_Filter for FuSa purposes. It does not exist on the GIC600\_Filter.
- GIC600AE\_Filter supports both Error Recovery Interrupt (ERI) and Fault Handling Interrupt (FHI).
- GIC600AE\_Filter has the limitation that it only supports error injection through the FMU\_SMINJERR register.

## Iris and MTI instances for GIC600AE\_Filter

This model has the following Iris instances:

Name	Instance type
GIC600AE_Filter	GIC600AE
GIC600AE_Filter.GICV3_ProtocolChecker	GICv3ProtocolChecker
GIC600AE_Filter.ITS0	GICv3InterruptTranslationService
GIC600AE_Filter.ITS0.ExportTest.GIC600AE_Filter.ITS0.pvbus_m[0].pvbusmaster	PVBusMaster
GIC600AE_Filter.rd_0	GICv3RedistributorInternal
GIC600AE_Filter.rd_0_0	GICv3RedistributorInternal
GIC600AE_Filter.rd_0_0_0	GICv3RedistributorInternal
GIC600AE_Filter.rd_0_0_0_0	GICv3Redistributor
GIC600AE_Filter.rd_0_0_0_0.ExportTest.GIC600AE_Filter.rd_0_0_0_0.pvbus_m[0].pvbusmaster	PVBusMaster
GIC600AE_Filter.rd_t1	GICv3Distributor
GIC600AE_Filter.rd_t1.ExportTest.GIC600AE_Filter.rd_t1.pvbus_m[0].pvbusmaster	PVBusMaster

This model has the following MTI trace components:

Name	Component type
GIC600AE_Filter.GICV3_ProtocolChecker	GICv3ProtocolChecker
GIC600AE_Filter.ITS0	GICv3InterruptTranslationService
GIC600AE_Filter.ITS0.ExportTest.GIC600AE_Filter.ITS0.pvbus_m[0].pvbusmaster	PVBusMaster
GIC600AE_Filter.rd_0	GICv3RedistributorInternal
GIC600AE_Filter.rd_0_0	GICv3RedistributorInternal
GIC600AE_Filter.rd_0_0_0	GICv3RedistributorInternal
GIC600AE_Filter.rd_0_0_0_0	GICv3Redistributor
GIC600AE_Filter.rd_0_0_0_0.ExportTest.GIC600AE_Filter.rd_0_0_0_0.pvbus_m[0].pvbusmaster	PVBusMaster
GIC600AE_Filter.rd_tl	GICv3Distributor
GIC600AE_Filter.rd_tl.ExportTest.GIC600AE_Filter.rd_tl.pvbus_m[0].pvbusmaster	PVBusMaster

### Ports for GIC600AE\_Filter

Port	Direction	Protocol	Description
apb_bus	slave	PVBus	APB4 port to FMU, which will be connected from safety island for FuSa
chip_id	slave	Value	chip_id port used for multichip operation
cpu_active_s	slave	Signal	CPUActive pins.
err_int	master	Signal	RAS Error Recovery Interrupt
fault_int	master	Signal	RAS Fault Handling Interrupt
fmua_error_int	master	Signal	FuSa FMU error interrupt signal
fmua_fault_int	master	Signal	FuSa FMU fault interrupt signal
po_reset	slave	Signal	Reset.
ppi_in_0	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 0.
ppi_in_100	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 100.
ppi_in_101	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 101.
ppi_in_102	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 102.
ppi_in_103	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 103.
ppi_in_104	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 104.
ppi_in_105	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 105.
ppi_in_106	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 106.
ppi_in_107	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 107.
ppi_in_108	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 108.
ppi_in_109	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 109.
ppi_in_110	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 110.
ppi_in_111	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 111.
ppi_in_112	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 112.
ppi_in_113	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 113.
ppi_in_114	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 114.

Port	Direction	Protocol	Description
ppi_in_115	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 115.
ppi_in_116	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 116.
ppi_in_117	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 117.
ppi_in_118	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 118.
ppi_in_119	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 119.
ppi_in_11	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 11.
ppi_in_120	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 120.
ppi_in_121	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 121.
ppi_in_122	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 122.
ppi_in_123	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 123.
ppi_in_124	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 124.
ppi_in_125	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 125.
ppi_in_126	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 126.
ppi_in_127	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 127.
ppi_in_128	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 128.
ppi_in_129	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 129.
ppi_in_12	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 12.
ppi_in_130	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 130.
ppi_in_131	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 131.
ppi_in_132	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 132.
ppi_in_133	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 133.
ppi_in_134	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 134.
ppi_in_135	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 135.
ppi_in_136	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 136.
ppi_in_137	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 137.
ppi_in_138	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 138.
ppi_in_139	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 139.
ppi_in_13	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 13.
ppi_in_140	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 140.
ppi_in_141	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 141.
ppi_in_142	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 142.
ppi_in_143	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 143.
ppi_in_144	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 144.
ppi_in_145	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 145.
ppi_in_146	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 146.
ppi_in_147	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 147.
ppi_in_148	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 148.
ppi_in_149	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 149.
ppi_in_14	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 14.
ppi_in_150	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 150.

Port	Direction	Protocol	Description
ppi_in_151	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 151.
ppi_in_152	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 152.
ppi_in_153	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 153.
ppi_in_154	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 154.
ppi_in_155	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 155.
ppi_in_156	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 156.
ppi_in_157	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 157.
ppi_in_158	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 158.
ppi_in_159	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 159.
ppi_in_15	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 15.
ppi_in_160	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 160.
ppi_in_161	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 161.
ppi_in_162	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 162.
ppi_in_163	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 163.
ppi_in_164	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 164.
ppi_in_165	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 165.
ppi_in_166	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 166.
ppi_in_167	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 167.
ppi_in_168	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 168.
ppi_in_169	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 169.
ppi_in_16	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 16.
ppi_in_170	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 170.
ppi_in_171	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 171.
ppi_in_172	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 172.
ppi_in_173	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 173.
ppi_in_174	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 174.
ppi_in_175	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 175.
ppi_in_176	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 176.
ppi_in_177	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 177.
ppi_in_178	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 178.
ppi_in_179	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 179.
ppi_in_17	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 17.
ppi_in_180	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 180.
ppi_in_181	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 181.
ppi_in_182	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 182.
ppi_in_183	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 183.
ppi_in_184	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 184.
ppi_in_185	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 185.
ppi_in_186	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 186.
ppi_in_187	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 187.

Port	Direction	Protocol	Description
ppi_in_188	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 188.
ppi_in_189	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 189.
ppi_in_18	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 18.
ppi_in_190	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 190.
ppi_in_191	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 191.
ppi_in_192	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 192.
ppi_in_193	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 193.
ppi_in_194	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 194.
ppi_in_195	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 195.
ppi_in_196	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 196.
ppi_in_197	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 197.
ppi_in_198	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 198.
ppi_in_199	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 199.
ppi_in_19	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 19.
ppi_in_1	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 1.
ppi_in_200	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 200.
ppi_in_201	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 201.
ppi_in_202	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 202.
ppi_in_203	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 203.
ppi_in_204	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 204.
ppi_in_205	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 205.
ppi_in_206	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 206.
ppi_in_207	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 207.
ppi_in_208	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 208.
ppi_in_209	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 209.
ppi_in_20	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 20.
ppi_in_210	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 210.
ppi_in_211	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 211.
ppi_in_212	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 212.
ppi_in_213	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 213.
ppi_in_214	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 214.
ppi_in_215	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 215.
ppi_in_216	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 216.
ppi_in_217	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 217.
ppi_in_218	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 218.
ppi_in_219	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 219.
ppi_in_21	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 21.
ppi_in_220	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 220.
ppi_in_221	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 221.
ppi_in_222	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 222.

Port	Direction	Protocol	Description
ppi_in_223	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 223.
ppi_in_224	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 224.
ppi_in_225	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 225.
ppi_in_226	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 226.
ppi_in_227	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 227.
ppi_in_228	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 228.
ppi_in_229	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 229.
ppi_in_22	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 22.
ppi_in_230	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 230.
ppi_in_231	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 231.
ppi_in_232	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 232.
ppi_in_233	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 233.
ppi_in_234	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 234.
ppi_in_235	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 235.
ppi_in_236	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 236.
ppi_in_237	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 237.
ppi_in_238	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 238.
ppi_in_239	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 239.
ppi_in_23	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 23.
ppi_in_240	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 240.
ppi_in_241	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 241.
ppi_in_242	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 242.
ppi_in_243	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 243.
ppi_in_244	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 244.
ppi_in_245	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 245.
ppi_in_246	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 246.
ppi_in_247	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 247.
ppi_in_248	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 248.
ppi_in_249	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 249.
ppi_in_24	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 24.
ppi_in_250	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 250.
ppi_in_251	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 251.
ppi_in_252	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 252.
ppi_in_253	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 253.
ppi_in_254	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 254.
ppi_in_255	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 255.
ppi_in_25	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 25.
ppi_in_26	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 26.
ppi_in_27	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 27.
ppi_in_28	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 28.

Port	Direction	Protocol	Description
ppi_in_29	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 29.
ppi_in_2	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 2.
ppi_in_30	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 30.
ppi_in_31	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 31.
ppi_in_32	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 32.
ppi_in_33	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 33.
ppi_in_34	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 34.
ppi_in_35	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 35.
ppi_in_36	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 36.
ppi_in_37	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 37.
ppi_in_38	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 38.
ppi_in_39	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 39.
ppi_in_3	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 3.
ppi_in_40	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 40.
ppi_in_41	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 41.
ppi_in_42	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 42.
ppi_in_43	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 43.
ppi_in_44	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 44.
ppi_in_45	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 45.
ppi_in_46	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 46.
ppi_in_47	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 47.
ppi_in_48	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 48.
ppi_in_49	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 49.
ppi_in_4	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 4.
ppi_in_50	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 50.
ppi_in_51	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 51.
ppi_in_52	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 52.
ppi_in_53	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 53.
ppi_in_54	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 54.
ppi_in_55	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 55.
ppi_in_56	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 56.
ppi_in_57	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 57.
ppi_in_58	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 58.
ppi_in_59	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 59.
ppi_in_5	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 5.
ppi_in_60	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 60.
ppi_in_61	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 61.
ppi_in_62	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 62.
ppi_in_63	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 63.
ppi_in_64	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 64.



Port	Direction	Protocol	Description
ppi_in_65	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 65.
ppi_in_66	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 66.
ppi_in_67	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 67.
ppi_in_68	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 68.
ppi_in_69	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 69.
ppi_in_6	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 6.
ppi_in_70	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 70.
ppi_in_71	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 71.
ppi_in_72	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 72.
ppi_in_73	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 73.
ppi_in_74	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 74.
ppi_in_75	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 75.
ppi_in_76	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 76.
ppi_in_77	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 77.
ppi_in_78	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 78.
ppi_in_79	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 79.
ppi_in_7	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 7.
ppi_in_80	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 80.
ppi_in_81	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 81.
ppi_in_82	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 82.
ppi_in_83	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 83.
ppi_in_84	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 84.
ppi_in_85	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 85.
ppi_in_86	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 86.
ppi_in_87	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 87.
ppi_in_88	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 88.
ppi_in_89	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 89.
ppi_in_8	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 8.
ppi_in_90	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 90.
ppi_in_91	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 91.
ppi_in_92	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 92.
ppi_in_93	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 93.
ppi_in_94	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 94.
ppi_in_95	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 95.
ppi_in_96	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 96.
ppi_in_97	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 97.
ppi_in_98	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 98.
ppi_in_99	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 99.
ppi_in_9	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 9.
pvbus_filtermiss_m	master	PVBus	Memory bus out. Transactions not filtered by the component.

Port	Direction	Protocol	Description
pvbus_m	master	PVBus	Memory bus out: transactions generated by the IRI.
pvbus_s	slave	PVBus	Memory bus in: memory-mapped register accesses are accepted on this interface.
redistributor_m	master	GICv3Comms	Input from and output to CPU interface.
reset	slave	Signal	Reset.
spi_in	slave	Signal	Shared peripheral interrupts.
wake_request	master	Signal	Power management outputs.

## Parameters for GIC600AE\_Filter

### CPU-affinities

A comma separated list of dotted quads containing the affinities of all PEs connected to this IRI. If CPU-affinities-file is specified, this parameter is ignored.

Type: `string`

Default value: N/A

### CPU-affinities-file

A file containing comma separated list of dotted quads containing the affinities of all PEs connected to this IRI. If this parameter is specified, CPU-affinities parameter will be ignored even when it is given.

Type: `string`

Default value: N/A

### DS-behaviour

GICD\_CTLR.DS field behaviour

0 :RAZ/WI

1 :RAO/WI

2 :RW.

Type: `int`

Default value: 2

### IIDR

GICD\_IIDR, GICR\_IIDR and GITS\_IIDR value. Defaults to the latest revision.

Type: `uint32_t`

Default value: 0x0300543B

**ITS-ID-bits**

Number of interrupt bits supported by ITS.

Type: `uint8_t`

Default value: 16

**ITS-collection-ID-bits**

Number of collection bits supported by ITS (optional parameter, 0 => 16bits support and GITS\_TYPER.CIL=0).

Type: `uint8_t`

Default value: 8

**ITS-count**

Number of Interrupt Translation Services to be instantiated (0=none).

Type: `uint8_t`

Default value: 1

**ITS-device-bits**

Number of bits supported for ITS device IDs.

Type: `uint8_t`

Default value: 16

**PPI-count**

Selects the number of PPI available for each PE

**8**

id22-27,29,30

**12**

id 20-31

**16**

id 16-31.

Type: `int`

Default value: 16

**SPI-blocks**

Number of SPI blocks supported by the IRI, each block contains 32 SPIs.

Type: `int`

Default value: 30

**affinity-width**

A dotted quad indicating the bitwidth of fields at each affinity level.

Type: `string`

Default value: `"4.8.8.8"`

**chip-id**

Chip ID for multichip operation.

Type: `uint8_t`

Default value: 0

**chip-select-affinity-level**

Affinity level 2 or 3 can be used for chip select.

Type: `int`

Default value: 3

**clear-ISPENDR-bit-for-level-sensitive-interrupt-when-acknowledged**

If true, acknowledgement of a level sensitive interrupt clears the corresponding bit in the ISPENDR register.

Type: `bool`

Default value: `false`

**direct-lpi-support**

Enable support for LPI operations through GICR registers.

Type: `bool`

Default value: `false`

**enable-multichip-operation**

Enables multi-chip operation between Distributors in distributed GIC IRI.

Type: `bool`

Default value: `true`

**enabled**

Enable GICv3 functionality; when false the component is inactive.

Type: `bool`

Default value: `true`

### **gicp-allow-ns-reset**

If true, non-secure read/write access to GICP register is allowed at reset. Not allowed otherwise. This emulates `gicp_allow_ns` tie-off signal.

Type: `bool`

Default value: `true`

### **gict-allow-ns-reset**

If true, non-secure read/write access to GICT register is allowed at reset. Not allowed otherwise. This emulates `gict_allow_ns` tie-off signal.

Type: `bool`

Default value: `true`

### **max-cores-supported-by-GCI**

GCI can support 1-64 cores. Maximum supported cores in GCI is configurable by device.

Type: `int`

Default value: 64

### **max-pe-on-chip**

Maximum number of cores on any single chip. This will be used to identify the target chip and core for multichip operation.

Type: `int`

Default value: 4

### **multichip-threaded-dgi**

Enable sending of multichip DGI messages in a separate thread.

Type: `bool`

Default value: `true`

### **print-memory-map**

Print memory map to stdout.

Type: `bool`

Default value: `false`

**redistributor-group**

Redistributor grouping information with affinity as JSON :

```
{
  "0": [
    "0.0.0.0",
    "0.0.0.1"
  ],
  "1": [
    "0.0.1.0",
    "0.0.1.1"
  ]
}
```

where RD with quad 0.0.0.0 and 0.0.0.1 belongs to RD group 0. All the RDs belong to one group when this parameter is not given.

Type: `string`

Default value: `N/A`

**redistributor-group-file**

File path to redistributor grouping information with affinity as JSON.

The file uses the same format as `redistributor-group` parameter.

Type: `string`

Default value: `N/A`

**redistributor-power-managed-by-pwrr**

GIC600AE redistributor power management is done by updating GICR\_PWRR register.

Type: `bool`

Default value: `true`

**reg-base**

GIC600AE base address.

Type: `uint64_t`

Default value: `0x2c010000`

**reg-base-per-redistributor**

Base address for each redistributor in the form:

```
0.0.0.0=0x2c010000, 0.0.0.1=0x2c020000
```

All redistributors must be specified and this overrides the `reg-base` parameter (except that `reg-base` will still be used for the top-level redistributor). If `reg-base-per-redistributor-file` is specified, this parameter is ignored.

Type: `string`

Default value: `""`

## 3.191 GIC600\_Filter

Defined in `LISA/GIC600.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### About GIC600\_Filter

GIC600 and GIC600\_Filter are minimal models of an Arm GIC-600 Generic Interrupt Controller, suitable for single-chip or multichip systems, where multichip operation has been tested, with the following exceptions:

- LPI/LPI command is implemented but untested.
- Power operation is unimplemented.

They provide a simple configuration interface that allows designers to introduce GIC600-like functionality to their systems, while only implementing the architectural behavior, as defined by the GICv3 architecture.

All implementation-specific registers and functionality are implemented.

As with the other GIC components, there are two variants of the model with slightly different memory interfaces. Both GIC600 and GIC600\_Filter have a `pvbus_s` port for register accesses and a `pvbus_m` port for the LPI-related traffic from redistributors and the ITS. The `pvbus_m` port is also used to compose multichip operation.

In addition, the GIC600\_Filter variant has a `pvbus_filtermiss_m` port, to which any transactions on the `pvbus_s` port and not directed to a 4K page used by the GIC are forwarded unmodified. Such transactions are terminated in the component when using the GIC600 variant.

It is recommended to use the GIC600 variant in most cases.

Also, the RAS register is implemented and various RAS errors are reported through status registers. Not all the RAS errors are available because Fast Models does not model errors that can cause ECC errors. Fast Models supports error record 0 and error record 13+.



- Set the `FASTSIM_GIC_MEMORY_MAP` environment variable to 1 to print to stderr the memory map of any GICv3 or later models that are included in the platform being run.
- For writes to `GITS_TRANSLATE64R`, the `MSIRewriter` component can be used to create the correct data transactions based on the device ID. For more details about `GITS_TRANSLATE64R`, see [MSIRewriter](#).

## ManagerID, ExtendedID, and UserFlags

This table shows how this model encodes information in the `ManagerID`, `ExtendedID`, and `UserFlags` bus attributes:

**Table 3-664: ManagerID, ExtendedID, and UserFlags**

PVBus attribute	Bits used	Property encoded	Notes
ManagerID	63:0	Manager ID that invoked the register access	–
ExtendedID	–	–	Not used
UserFlags	–	–	Not used



The `pdbus_m` port does not use `ManagerID`, `ExtendedID`, or `UserFlags`.

## Iris and MTI instances for GIC600\_Filter

This model has the following Iris instances:

Name	Instance type
<code>GIC600_Filter</code>	<code>GIC600</code>
<code>GIC600_Filter.GICV3_ProtocolChecker</code>	<code>GICv3ProtocolChecker</code>
<code>GIC600_Filter.ITS0</code>	<code>GICv3InterruptTranslationService</code>
<code>GIC600_Filter.ITS0.ExportTest.GIC600_Filter.ITS0.pdbus_m[0].pdbusmaster</code>	<code>PVBusMaster</code>
<code>GIC600_Filter.rd_0</code>	<code>GICv3RedistributorInternal</code>
<code>GIC600_Filter.rd_0_0</code>	<code>GICv3RedistributorInternal</code>
<code>GIC600_Filter.rd_0_0_0</code>	<code>GICv3RedistributorInternal</code>
<code>GIC600_Filter.rd_0_0_0_0</code>	<code>GICv3Redistributor</code>
<code>GIC600_Filter.rd_0_0_0_0.ExportTest.GIC600_Filter.rd_0_0_0_0.pdbus_m[0].pdbusmaster</code>	<code>PVBusMaster</code>
<code>GIC600_Filter.rd_t1</code>	<code>GICv3Distributor</code>
<code>GIC600_Filter.rd_t1.ExportTest.GIC600_Filter.rd_t1.pdbus_m[0].pdbusmaster</code>	<code>PVBusMaster</code>

This model has the following MTI trace components:

Name	Component type
<code>GIC600_Filter.GICV3_ProtocolChecker</code>	<code>GICv3ProtocolChecker</code>



Name	Component type
GIC600_Filter.ITS0	GICv3InterruptTranslationService
GIC600_Filter.ITS0.ExportTest.GIC600_Filter.ITS0.pvbus_m[0].pvbusmaster	PVBusMaster
GIC600_Filter.rd_0	GICv3RedistributorInternal
GIC600_Filter.rd_0_0	GICv3RedistributorInternal
GIC600_Filter.rd_0_0_0	GICv3RedistributorInternal
GIC600_Filter.rd_0_0_0_0	GICv3Redistributor
GIC600_Filter.rd_0_0_0_0.ExportTest.GIC600_Filter.rd_0_0_0_0.pvbus_m[0].pvbusmaster	PVBusMaster
GIC600_Filter.rd_tl	GICv3Distributor
GIC600_Filter.rd_tl.ExportTest.GIC600_Filter.rd_tl.pvbus_m[0].pvbusmaster	PVBusMaster

### Ports for GIC600\_Filter

Port	Direction	Protocol	Description
chip_id	slave	Value	chip_id port which is valid from GIC600 r1p2. Writing to this port for prior GIC600 version will be ignored.
cpu_active_s	slave	Signal	CPUActive pins.
po_reset	slave	Signal	Reset.
ppi_in_0	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 0.
ppi_in_100	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 100.
ppi_in_101	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 101.
ppi_in_102	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 102.
ppi_in_103	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 103.
ppi_in_104	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 104.
ppi_in_105	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 105.
ppi_in_106	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 106.
ppi_in_107	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 107.
ppi_in_108	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 108.
ppi_in_109	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 109.
ppi_in_10	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 10.
ppi_in_110	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 110.
ppi_in_111	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 111.
ppi_in_112	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 112.
ppi_in_113	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 113.
ppi_in_114	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 114.
ppi_in_115	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 115.
ppi_in_116	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 116.
ppi_in_117	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 117.
ppi_in_118	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 118.
ppi_in_119	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 119.
ppi_in_11	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 11.
ppi_in_120	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 120.

Port	Direction	Protocol	Description
ppi_in_121	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 121.
ppi_in_122	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 122.
ppi_in_123	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 123.
ppi_in_124	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 124.
ppi_in_125	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 125.
ppi_in_126	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 126.
ppi_in_127	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 127.
ppi_in_128	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 128.
ppi_in_129	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 129.
ppi_in_12	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 12.
ppi_in_130	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 130.
ppi_in_131	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 131.
ppi_in_132	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 132.
ppi_in_133	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 133.
ppi_in_134	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 134.
ppi_in_135	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 135.
ppi_in_136	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 136.
ppi_in_137	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 137.
ppi_in_138	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 138.
ppi_in_139	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 139.
ppi_in_13	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 13.
ppi_in_140	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 140.
ppi_in_141	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 141.
ppi_in_142	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 142.
ppi_in_143	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 143.
ppi_in_144	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 144.
ppi_in_145	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 145.
ppi_in_146	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 146.
ppi_in_147	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 147.
ppi_in_148	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 148.
ppi_in_149	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 149.
ppi_in_14	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 14.
ppi_in_150	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 150.
ppi_in_151	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 151.
ppi_in_152	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 152.
ppi_in_153	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 153.
ppi_in_154	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 154.
ppi_in_155	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 155.
ppi_in_156	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 156.
ppi_in_157	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 157.

Port	Direction	Protocol	Description
ppi_in_158	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 158.
ppi_in_159	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 159.
ppi_in_15	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 15.
ppi_in_160	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 160.
ppi_in_161	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 161.
ppi_in_162	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 162.
ppi_in_163	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 163.
ppi_in_164	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 164.
ppi_in_165	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 165.
ppi_in_166	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 166.
ppi_in_167	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 167.
ppi_in_168	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 168.
ppi_in_169	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 169.
ppi_in_16	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 16.
ppi_in_170	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 170.
ppi_in_171	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 171.
ppi_in_172	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 172.
ppi_in_173	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 173.
ppi_in_174	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 174.
ppi_in_175	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 175.
ppi_in_176	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 176.
ppi_in_177	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 177.
ppi_in_178	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 178.
ppi_in_179	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 179.
ppi_in_17	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 17.
ppi_in_180	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 180.
ppi_in_181	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 181.
ppi_in_182	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 182.
ppi_in_183	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 183.
ppi_in_184	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 184.
ppi_in_185	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 185.
ppi_in_186	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 186.
ppi_in_187	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 187.
ppi_in_188	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 188.
ppi_in_189	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 189.
ppi_in_18	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 18.
ppi_in_190	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 190.
ppi_in_191	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 191.
ppi_in_192	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 192.
ppi_in_193	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 193.

Port	Direction	Protocol	Description
ppi_in_194	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 194.
ppi_in_195	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 195.
ppi_in_196	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 196.
ppi_in_197	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 197.
ppi_in_198	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 198.
ppi_in_199	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 199.
ppi_in_19	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 19.
ppi_in_1	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 1.
ppi_in_200	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 200.
ppi_in_201	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 201.
ppi_in_202	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 202.
ppi_in_203	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 203.
ppi_in_204	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 204.
ppi_in_205	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 205.
ppi_in_206	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 206.
ppi_in_207	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 207.
ppi_in_208	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 208.
ppi_in_209	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 209.
ppi_in_20	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 20.
ppi_in_210	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 210.
ppi_in_211	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 211.
ppi_in_212	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 212.
ppi_in_213	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 213.
ppi_in_214	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 214.
ppi_in_215	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 215.
ppi_in_216	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 216.
ppi_in_217	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 217.
ppi_in_218	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 218.
ppi_in_219	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 219.
ppi_in_21	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 21.
ppi_in_220	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 220.
ppi_in_221	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 221.
ppi_in_222	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 222.
ppi_in_223	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 223.
ppi_in_224	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 224.
ppi_in_225	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 225.
ppi_in_226	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 226.
ppi_in_227	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 227.
ppi_in_228	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 228.
ppi_in_229	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 229.

Port	Direction	Protocol	Description
ppi_in_22	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 22.
ppi_in_230	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 230.
ppi_in_231	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 231.
ppi_in_232	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 232.
ppi_in_233	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 233.
ppi_in_234	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 234.
ppi_in_235	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 235.
ppi_in_236	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 236.
ppi_in_237	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 237.
ppi_in_238	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 238.
ppi_in_239	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 239.
ppi_in_23	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 23.
ppi_in_240	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 240.
ppi_in_241	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 241.
ppi_in_242	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 242.
ppi_in_243	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 243.
ppi_in_244	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 244.
ppi_in_245	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 245.
ppi_in_246	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 246.
ppi_in_247	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 247.
ppi_in_248	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 248.
ppi_in_249	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 249.
ppi_in_24	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 24.
ppi_in_250	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 250.
ppi_in_251	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 251.
ppi_in_252	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 252.
ppi_in_253	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 253.
ppi_in_254	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 254.
ppi_in_255	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 255.
ppi_in_25	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 25.
ppi_in_26	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 26.
ppi_in_27	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 27.
ppi_in_28	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 28.
ppi_in_29	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 29.
ppi_in_2	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 2.
ppi_in_30	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 30.
ppi_in_31	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 31.
ppi_in_32	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 32.
ppi_in_33	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 33.
ppi_in_34	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 34.

Port	Direction	Protocol	Description
ppi_in_35	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 35.
ppi_in_36	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 36.
ppi_in_37	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 37.
ppi_in_38	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 38.
ppi_in_39	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 39.
ppi_in_3	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 3.
ppi_in_40	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 40.
ppi_in_41	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 41.
ppi_in_42	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 42.
ppi_in_43	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 43.
ppi_in_44	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 44.
ppi_in_45	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 45.
ppi_in_46	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 46.
ppi_in_47	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 47.
ppi_in_48	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 48.
ppi_in_49	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 49.
ppi_in_4	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 4.
ppi_in_50	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 50.
ppi_in_51	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 51.
ppi_in_52	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 52.
ppi_in_53	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 53.
ppi_in_54	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 54.
ppi_in_55	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 55.
ppi_in_56	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 56.
ppi_in_57	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 57.
ppi_in_58	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 58.
ppi_in_59	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 59.
ppi_in_5	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 5.
ppi_in_60	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 60.
ppi_in_61	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 61.
ppi_in_62	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 62.
ppi_in_63	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 63.
ppi_in_64	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 64.
ppi_in_65	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 65.
ppi_in_66	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 66.
ppi_in_67	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 67.
ppi_in_68	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 68.
ppi_in_69	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 69.
ppi_in_6	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 6.
ppi_in_70	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 70.

Port	Direction	Protocol	Description
ppi_in_71	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 71.
ppi_in_72	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 72.
ppi_in_73	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 73.
ppi_in_74	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 74.
ppi_in_75	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 75.
ppi_in_76	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 76.
ppi_in_77	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 77.
ppi_in_78	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 78.
ppi_in_79	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 79.
ppi_in_7	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 7.
ppi_in_80	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 80.
ppi_in_81	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 81.
ppi_in_82	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 82.
ppi_in_83	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 83.
ppi_in_84	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 84.
ppi_in_85	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 85.
ppi_in_86	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 86.
ppi_in_87	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 87.
ppi_in_88	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 88.
ppi_in_89	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 89.
ppi_in_8	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 8.
ppi_in_90	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 90.
ppi_in_91	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 91.
ppi_in_92	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 92.
ppi_in_93	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 93.
ppi_in_94	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 94.
ppi_in_95	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 95.
ppi_in_96	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 96.
ppi_in_97	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 97.
ppi_in_98	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 98.
ppi_in_99	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 99.
ppi_in_9	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 9.
pvbus_filtermiss_m	master	PVBus	Memory bus out. Transactions not filtered by the component.
pvbus_m	master	PVBus	Memory bus out: transactions generated by the IRI.
pvbus_s	slave	PVBus	Memory bus in: memory-mapped register accesses are accepted on this interface.
redistributor_m	master	GLCv3Comms	Input from and output to CPU interface.
reset	slave	Signal	Reset.
spi_in	slave	Signal	Shared peripheral interrupts.
wake_request	master	Signal	Power management outputs.

## Parameters for GIC600\_Filter

### **CPU-affinities**

A comma separated list of dotted quads containing the affinities of all PEs connected to this IRI. If CPU-affinities-file is specified, this parameter is ignored.

Type: `string`

Default value: `N/A`

### **CPU-affinities-file**

A file containing comma separated list of dotted quads containing the affinities of all PEs connected to this IRI. If this parameter is specified, CPU-affinities parameter will be ignored even when it is given.

Type: `string`

Default value: `N/A`

### **DS-behaviour**

GICD\_CTLR.DS field behaviour

0 : **RAZ/WI**

1 : **RAO/WI**

2 : **RW**.

Type: `int`

Default value: 2

### **IIDR**

GICD\_IIDR, GICR\_IIDR and GITS\_IIDR value. Defaults to the latest revision.

Type: `uint32_t`

Default value: `0x0201743B`

### **ITS-ID-bits**

Number of interrupt bits supported by ITS.

Type: `uint8_t`

Default value: 16



**ITS-collection-ID-bits**

Number of collection bits supported by ITS (optional parameter, 0 => 16bits support and GITS\_TYPER.CIL=0).

Type: `uint8_t`

Default value: 8

**ITS-count**

Number of Interrupt Translation Services to be instantiated (0=none).

Type: `uint8_t`

Default value: 1

**ITS-device-bits**

Number of bits supported for ITS device IDs.

Type: `uint8_t`

Default value: 16

**PPI-count**

Selects the number of PPI available for each PE

8 :id22-27,29,30

**12**

id 20-31

**16**

id 16-31.

Type: `int`

Default value: 16

**RAS-CFI-support**

If true, fault handling interrupt for corrected errors is supported. Not supported otherwise.

Type: `bool`

Default value: `false`

**RAS-FI-support**

If true, fault handling interrupt is supported. Not supported otherwise.

Type: `bool`

Default value: `false`

### **RAS-UE-support**

If true, In-band uncorrected error reporting is supported. Not supported otherwise.

Type: `bool`

Default value: `false`

### **RAS-UI-support**

If true, error recovery interrupt for uncorrected errors is supported. Not supported otherwise.

Type: `bool`

Default value: `false`

### **SPI-blocks**

Number of SPI blocks supported by the IRI, each block contains 32 SPIs.

Type: `int`

Default value: 30

### **affinity-width**

A dotted quad indicating the bitwidth of fields at each affinity level.

Type: `string`

Default value: `"4.8.8.8"`

### **chip-id**

Chip ID when multichip operation is enabled.

Type: `uint8_t`

Default value: 0

### **chip-select-affinity-level**

Affinity level 2 or 3 can be used for chip select.

Type: `int`

Default value: 3

### **clear-ISPENDR-bit-for-level-sensitive-interrupt-when-acknowledged**

If true, acknowledgement of a level sensitive interrupt clears the corresponding bit in the ISPENDR register.

Type: `bool`

Default value: `false`

### **direct-lpi-support**

Enable support for LPI operations through GICR registers.

Type: `bool`

Default value: `false`

### **enable-multichip-operation**

Enables multi-chip operation between Distributors in distributed GIC IRI.

Type: `bool`

Default value: `true`

### **enabled**

Enable GICv3 functionality; when false the component is inactive.

Type: `bool`

Default value: `true`

### **gicp-allow-ns-reset**

If true, non-secure read/write access to GICP register is allowed at reset. Not allowed otherwise. This emulates `gicp_allow_ns` tie-off signal.

Type: `bool`

Default value: `true`

### **gict-allow-ns-reset**

If true, non-secure read/write access to GICT register is allowed at reset. Not allowed otherwise. This emulates `gict_allow_ns` tie-off signal.

Type: `bool`

Default value: `true`

### **max-cores-supported-by-GCI**

GCI can support 1-64 cores. Maximum supported cores in GCI is configurable by device.

Type: `int`

Default value: `64`

**max-pe-on-chip**

Maximum number of cores on any single chip. This will be used to identify the target chip and core when multichip operation is enabled.

Type: `int`

Default value: `4`

**multichip-threaded-dgi**

Enable sending of multichip DGI messages in a separate thread, when multichip operation is enabled.

Type: `bool`

Default value: `true`

**print-memory-map**

Print memory map to stdout.

Type: `bool`

Default value: `false`

**redistributor-group**

Redistributor grouping information with affinity as JSON :

```
{
  "0": [
    "0.0.0.0",
    "0.0.0.1"
  ],
  "1": [
    "0.0.1.0",
    "0.0.1.1"
  ]
}
```

where RD with quad 0.0.0.0 and 0.0.0.1 belongs to RD group 0. All the RDs belong to one group when this parameter is not given.

Type: `string`

Default value: `N/A`

**redistributor-group-file**

File path to redistributor grouping information with affinity as JSON.

The file uses the same format as `redistributor-group` parameter.

Type: `string`

Default value: `N/A`

**redistributor-power-managed-by-pwrr**

GIC600 redistributor power management is done by updating GICR\_PWRR register.

Type: `bool`  
  
Default value: `true`

**reg-base**

GIC-600 base address.

Type: `uint64_t`  
  
Default value: `0x2c010000`

**reg-base-per-redistributor**

Base address for each redistributor in the form:

```
0.0.0.0=0x2c010000, 0.0.0.1=0x2c020000
```

All redistributors must be specified and this overrides the `reg-base` parameter (except that `reg-base` will still be used for the top-level redistributor). If `reg-base-per-redistributor-file` is specified, this parameter is ignored.

Type: `string`  
  
Default value: `""`

3.192 GIC625

Defined in `LISA/GIC625.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p1	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

About GIC625

GIC625 and GIC625\_Filter are Generic Interrupt Controllers that handle interrupts from peripherals to cores and interrupts between cores in a single cluster of up to 8 cores. They support the GICv3 and GICv3.1 architectures.

GIC625 supports the following features:

- Grouping cores in GCI.

- LPI is disabled.
- Power Management updates such as:
  - GCI grouping.
  - GCI RDPowerDown support
  - Updating the GCI ID for the cluster.
  - GCI GICR\_PWRR register updates.
- Inversion of SPI and PPI inputs.
- Extended PPI.



- Extended SPI is not supported.
- All implementation-specific registers and functionality are implemented.

## Iris and MTI instances for GIC625

This model has the following Iris instances:

Name	Instance type
GIC625	GIC625
GIC625.GICV3_ProtocolChecker	GICv3ProtocolChecker
GIC625.rd_0	GICv3RedistributorInternal
GIC625.rd_0_0	GICv3RedistributorInternal
GIC625.rd_0_0_0	GICv3RedistributorInternal
GIC625.rd_0_0_0_0	GICv3Redistributor
GIC625.rd_0_0_0_0.ExportTest.GIC625.rd_0_0_0_0.pvbus_m[0].pvbusmaster	PVBusMaster
GIC625.rd_t1	GICv3Distributor

This model has the following MTI trace components:

Name	Component type
GIC625.GICV3_ProtocolChecker	GICv3ProtocolChecker
GIC625.rd_0	GICv3RedistributorInternal
GIC625.rd_0_0	GICv3RedistributorInternal
GIC625.rd_0_0_0	GICv3RedistributorInternal
GIC625.rd_0_0_0_0	GICv3Redistributor
GIC625.rd_0_0_0_0.ExportTest.GIC625.rd_0_0_0_0.pvbus_m[0].pvbusmaster	PVBusMaster
GIC625.rd_t1	GICv3Distributor

## Ports for GIC625

Port	Direction	Protocol	Description
chip_id	slave	Value	chip_id port used for multichip operation

Port	Direction	Protocol	Description
cpu_active_s	slave	Signal	CPUActive pins.
po_reset	slave	Signal	Resets. This is used as a dbg_reset in TRM, and also be used for cold reset for PMU and FMU.
ppi_in_0	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 0.
ppi_in_1	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 1.
ppi_in_2	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 2.
ppi_in_3	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 3.
ppi_in_4	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 4.
ppi_in_5	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 5.
ppi_in_6	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 6.
ppi_in_7	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 7.
pvbuss_m	master	PVBus	Memory bus out: transactions generated by the IRI.
pvbuss_s	slave	PVBus	Memory bus in: memory-mapped register accesses are accepted on this interface.
redistributor_m	master	GICv3Comms	Input from and output to CPU interface.
reset	slave	Signal	Resets.
spi_in	slave	Signal	Shared peripheral interrupts.
wake_request	master	Signal	Power management outputs.

## Parameters for GIC625

### CPU-affinities

A comma separated list of dotted quads containing the affinities of all PEs connected to this IRI. If CPU-affinities-file is specified, this parameter is ignored.

Type: string

Default value: N/A

### CPU-affinities-file

A file containing comma separated list of dotted quads containing the affinities of all PEs connected to this IRI. If this parameter is specified, CPU-affinities parameter will be ignored even when it is given.

Type: string

Default value: N/A

### IIDR

GICD\_IIDR, GICR\_IIDR and GITS\_IIDR value. Defaults to the latest revision.

Type: uint32\_t

Default value: 0x0601043B

**PPI-count**

Selects the number of PPI available for each PE

**8**

id22-27,29,30

**12**

id 20-31

**16**

id 16-31.

Type: `int`

Default value: 16

**SPI-blocks**

Number of SPI blocks supported by the IRI, each block contains 32 SPIs.

Type: `int`

Default value: 30

**affinity-width**

A dotted quad indicating the bitwidth of fields at each affinity level.

Type: `string`

Default value: "4.8.8.8"

**chip-select-affinity-level**

Affinity level 2 or 3 can be used for chip select.

Type: `int`

Default value: 3

**clear-ISPENDR-bit-for-level-sensitive-interrupt-when-acknowledged**

If true, acknowledgement of a level sensitive interrupt clears the corresponding bit in the ISPENDR register.

Type: `bool`

Default value: `false`

**enabled**

Enable GICv3 functionality; when false the component is inactive.

Type: `bool`



Default value: `true`

### **gicp-allow-ns-reset**

If true, non-secure read/write access to GICP register is allowed at reset. Not allowed otherwise. This emulates `gicp_allow_ns` tie-off signal.

Type: `bool`

Default value: `true`

### **gict-allow-ns-reset**

If true, non-secure read/write access to GICT register is allowed at reset. Not allowed otherwise. This emulates `gict_allow_ns` tie-off signal.

Type: `bool`

Default value: `true`

### **has-two-security-states**

If true, has two security states i.e. `GICD_CTLR.DS=0`. This emulates `gicd_ctrl_ds` tie-off signal.

Type: `bool`

Default value: `true`

### **max-cores-supported-by-GCI**

GCI can support 1-8 cores. Maximum supported cores in GCI is configurable by device.

Type: `int`

Default value: 8

### **print-memory-map**

Print memory map to stdout.

Type: `bool`

Default value: `false`

### **redistributor-group**

Redistributor grouping information with affinity as JSON :

```
{
  "0": [
    "0.0.0.0",
    "0.0.0.1"
  ],
  "1": [
    "0.0.1.0",
    "0.0.1.1"
  ]
}
```

```
}
```

where RD with quad 0.0.0.0 and 0.0.0.1 belongs to RD group 0. All the RDs belong to one group when this parameter is not given.

Type: `string`

Default value: `N/A`

### **redistributor-group-file**

File path to redistributor grouping information with affinity as JSON.

The file uses the same format as `redistributor-group` parameter.

Type: `string`

Default value: `N/A`

### **redistributor-power-managed-by-pwrr**

GIC625 redistributor power management is done by updating GICR\_PWRR register.

Type: `bool`

Default value: `true`

### **reg-base**

GIC625 base address.

Type: `uint64_t`

Default value: `0x2c010000`

### **reg-base-per-redistributor**

Base address for each redistributor in the form:

```
0.0.0.0=0x2c010000, 0.0.0.1=0x2c020000
```

All redistributors must be specified and this overrides the `reg-base` parameter (except that `reg-base` will still be used for the top-level redistributor). If `reg-base-per-redistributor-file` is specified, this parameter is ignored.

Type: `string`

Default value: `""`

## 3.193 GIC625\_Filter

Defined in `LISA/GIC625.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### About GIC625\_Filter

GIC625 and GIC625\_Filter are Generic Interrupt Controllers that handle interrupts from peripherals to cores and interrupts between cores in a single cluster of up to 8 cores. They support the GICv3 and GICv3.1 architectures.

GIC625\_Filter supports the following features:

- Grouping cores in GCI.
- LPI is disabled.
- Power Management updates such as:
  - GCI grouping.
  - GCI RDPowerDown support
  - Updating the GCI ID for the cluster.
  - GCI GICR\_PWRR register updates.
- Inversion of SPI and PPI inputs.
- Extended PPI.



Note

- Extended SPI is not supported.
- All implementation-specific registers and functionality are implemented.

### Iris and MTI instances for GIC625\_Filter

This model has the following Iris instances:

Name	Instance type
GIC625_Filter	GIC625
GIC625_Filter.GICV3_ProtocolChecker	GICv3ProtocolChecker
GIC625_Filter.rd_0	GICv3RedistributorInternal
GIC625_Filter.rd_0_0	GICv3RedistributorInternal
GIC625_Filter.rd_0_0_0	GICv3RedistributorInternal
GIC625_Filter.rd_0_0_0_0	GICv3Redistributor

Name	Instance type
GIC625_Filter.rd_0_0_0_0.ExportTest.GIC625_Filter.rd_0_0_0_0.pvbus_m[0].pvbusmaster	PVBusMaster
GIC625_Filter.rd_tl	GICv3Distributor

This model has the following MTI trace components:

Name	Component type
GIC625_Filter.GICV3_ProtocolChecker	GICv3ProtocolChecker
GIC625_Filter.rd_0	GICv3RedistributorInternal
GIC625_Filter.rd_0_0	GICv3RedistributorInternal
GIC625_Filter.rd_0_0_0	GICv3RedistributorInternal
GIC625_Filter.rd_0_0_0_0	GICv3Redistributor
GIC625_Filter.rd_0_0_0_0.ExportTest.GIC625_Filter.rd_0_0_0_0.pvbus_m[0].pvbusmaster	PVBusMaster
GIC625_Filter.rd_tl	GICv3Distributor

### Ports for GIC625\_Filter

Port	Direction	Protocol	Description
chip_id	slave	Value	chip_id port used for multichip operation
cpu_active_s	slave	Signal	CPUActive pins.
po_reset	slave	Signal	Reset.
ppi_in_0	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 0.
ppi_in_1	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 1.
ppi_in_2	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 2.
ppi_in_3	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 3.
ppi_in_4	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 4.
ppi_in_5	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 5.
ppi_in_6	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 6.
ppi_in_7	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 7.
pvbus_filtermiss_m	master	PVBus	Memory bus out. Transactions not filtered by the component.
pvbus_m	master	PVBus	Memory bus out: transactions generated by the IRI.
pvbus_s	slave	PVBus	Memory bus in: memory-mapped register accesses are accepted on this interface.
redistributor_m	master	GICv3Comms	Input from and output to CPU interface.
reset	slave	Signal	Reset.
spi_in	slave	Signal	Shared peripheral interrupts.
wake_request	master	Signal	Power management outputs.

### Parameters for GIC625\_Filter

#### CPU-affinities

A comma separated list of dotted quads containing the affinities of all PEs connected to this IRI. If CPU-affinities-file is specified, this parameter is ignored.

Type: `string`

Default value: N/A

### **CPU-affinities-file**

A file containing comma separated list of dotted quads containing the affinities of all PEs connected to this IRI. If this parameter is specified, CPU-affinities parameter will be ignored even when it is given.

Type: `string`

Default value: N/A

### **IIDR**

GICD\_IIDR, GICR\_IIDR and GITS\_IIDR value. Defaults to the latest revision.

Type: `uint32_t`

Default value: 0x0601043B

### **PPI-count**

Selects the number of PPI available for each PE

**8**

id22-27,29,30

**12**

id 20-31

**16**

id 16-31.

Type: `int`

Default value: 16

### **SPI-blocks**

Number of SPI blocks supported by the IRI, each block contains 32 SPIs.

Type: `int`

Default value: 30

### **affinity-width**

A dotted quad indicating the bitwidth of fields at each affinity level.

Type: `string`

Default value: "4.8.8.8"

**chip-select-affinity-level**

Affinity level 2 or 3 can be used for chip select.

Type: `int`

Default value: `3`

**clear-ISPENDR-bit-for-level-sensitive-interrupt-when-acknowledged**

If true, acknowledgement of a level sensitive interrupt clears the corresponding bit in the ISPENDR register.

Type: `bool`

Default value: `false`

**enabled**

Enable GICv3 functionality; when false the component is inactive.

Type: `bool`

Default value: `true`

**gicp-allow-ns-reset**

If true, non-secure read/write access to GICP register is allowed at reset. Not allowed otherwise. This emulates `gicp_allow_ns` tie-off signal.

Type: `bool`

Default value: `true`

**gict-allow-ns-reset**

If true, non-secure read/write access to GICT register is allowed at reset. Not allowed otherwise. This emulates `gict_allow_ns` tie-off signal.

Type: `bool`

Default value: `true`

**has-two-security-states**

If true, has two security states i.e. `GICD_CTLR.DS=0`. This emulates `gicd_ctlr_ds` tie-off signal.

Type: `bool`

Default value: `true`

**max-cores-supported-by-GCI**

GCI can support 8 cores. Maximum supported cores in GCI is configurable by device.

Type: `int`

Default value: `8`

### **print-memory-map**

Print memory map to stdout.

Type: `bool`

Default value: `false`

### **redistributor-group**

Redistributor grouping information with affinity as JSON :

```
{
  "0": [
    "0.0.0.0",
    "0.0.0.1"
  ],
  "1": [
    "0.0.1.0",
    "0.0.1.1"
  ]
}
```

where RD with quad 0.0.0.0 and 0.0.0.1 belongs to RD group 0. All the RDs belong to one group when this parameter is not given.

Type: `string`

Default value: `N/A`

### **redistributor-group-file**

File path to redistributor grouping information with affinity as JSON.

The file uses the same format as `redistributor-group` parameter.

Type: `string`

Default value: `N/A`

### **redistributor-power-managed-by-pwrr**

GIC625 redistributor power management is done by updating GICR\_PWRR register.

Type: `bool`

Default value: `true`

### **reg-base**

GIC625 base address.

Type: `uint64_t`

Default value: 0x2c010000

### **reg-base-per-redistributor**

Base address for each redistributor in the form:

```
0.0.0.0=0x2c010000, 0.0.0.1=0x2c020000
```

All redistributors must be specified and this overrides the reg-base parameter (except that reg-base will still be used for the top-level redistributor). If `reg-base-per-redistributor-file` is specified, this parameter is ignored.

Type: `string`

Default value: ""

## 3.194 GIC700

Defined in `LISA/GIC700.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support
r1p0	Full support
r2p0	Full support
r3p0	Full support
r4p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### About GIC700

GIC700 and GIC700\_Filter are minimal models of an Arm® GIC-700 Generic Interrupt Controller, suitable for single-chip or multichip systems. They support the following multichip operations:

- Multichip configuration through distributor registers
- SPI
- SGI
- Physical LPI
- Physical LPI command
- Virtual LPI
- Virtual LPI command



The models provide a simple configuration interface that allows designers to introduce GIC700-like functionality to systems, while only implementing the architectural behavior, as defined by the GICv3/GICv4 architecture.

All implementation-specific registers and functionality are implemented.

As with the other GIC components, there are two variants of the model with slightly different memory interfaces. Both GIC700 and GIC700\_Filter have a `pvbuss_s` port for register accesses and a `pvbuss_m` port for the LPI-related traffic from redistributors and the ITS.

In addition, the GIC700\_Filter variant has a `pvbuss_filtermiss_m` port, to which any transactions on the `pvbuss_s` port and not directed to a 4 KB page used by the GIC are forwarded unmodified. Such transactions are terminated in the component when using the GIC700 variant.

GIC700 and GIC700\_Filter support an extended interrupt range, with 64 more PPIs and 1024 more SPIs. To program additional interrupts, refer to the GICv3/GICv4 specification.

GIC700 and GIC700\_Filter support direct injection of virtual interrupts. This reduces the overhead of trapping to hypervisor for the generation of virtual interrupts. In the GICv3/GICv4 specification, direct injection is supported in two different versions, GICv4.0 and GICv4.1. GIC700 and GIC700\_Filter support GICv4.1.

It is recommended to use GIC700 instead of GIC700\_Filter in most cases.

The GIC700 model supports multiple revisions of the IP. By default, its behavior is set to the latest revision supported. Parameters can be used to enable or disable features that are specific to earlier revisions. The LCA (Local Cross-chip Addressing) feature which is added in r2p0 is disabled by default and is enabled by setting the parameter `enable-local-cross-chip-addressing` to true. The MultiView feature which is added in r4p0 is disabled by default and is enabled by setting the parameter `enable-multiple-views-feature` to true. The model's registers follow the specification for the latest IP revision supported.



- Set the `FASTSIM_GIC_MEMORY_MAP` environment variable to 1 to print to stderr the memory map of any GICv3 or later models that are included in the platform being run.
  - For writes to `GITS_TRANSLATE64R`, the MSIRewriter component can be used to create the correct data transactions based on the device ID. For more details about `GITS_TRANSLATE64R`, see [MSIRewriter](#).
- 

## Limitations

- The following multichip operations are not yet supported:
  - vPE control.
  - Power control.
- There is no support for the VMOV command.

- Limited testing has been performed for multichip LPI operations, so the model might contain defects.
- One ITS is supported for each GIC for multichip operations.
- There is no support for VICM.
- There is no support for PMU.
- There is no support for real time interrupt.
- Multiview feature for ITS is not supported.
- Assigning same affinity to different views is not supported.

### ManagerID, ExtendedID, and UserFlags

This table shows how this model encodes information in the `ManagerID`, `ExtendedID`, and `UserFlags` bus attributes:

**Table 3-677: ManagerID, ExtendedID, and UserFlags**

PVBus attribute	Bits used	Property encoded	Notes
ManagerID	63:0	Manager ID that invoked the register access	–
ExtendedID	–	–	Not used
UserFlags	–	–	Not used



The `pvbus_m` port does not use `ManagerID`, `ExtendedID`, or `UserFlags`.

### Iris and MTI instances for GIC700

This model has the following Iris instances:

Name	Instance type
GIC700	GIC700
GIC700.GICV3_ProtocolChecker	GICv3ProtocolChecker
GIC700.ITS0	GICv4InterruptTranslationService
GIC700.ITS0.ExportTest.GIC700.ITS0.pvbus_m[0].pvbusmaster	PVBusMaster
GIC700.ITS0.bus_subordinate	PVBusSlave
GIC700.rd_0	GICv4RedistributorInternal
GIC700.rd_0_0	GICv4RedistributorInternal
GIC700.rd_0_0_0	GICv4RedistributorInternal
GIC700.rd_0_0_0_0	GICv3Redistributor
GIC700.rd_0_0_0_0.ExportTest.GIC700.rd_0_0_0_0.pvbus_m[0].pvbusmaster	PVBusMaster
GIC700.rd_t1	GICv3Distributor
GIC700.rd_t1.ExportTest.GIC700.rd_t1.pvbus_m[0].pvbusmaster	PVBusMaster

This model has the following MTI trace components:

Name	Component type
GIC700.GICV3_ProtocolChecker	GICv3ProtocolChecker
GIC700.ITS0	GICv4InterruptTranslationService
GIC700.ITS0.ExportTest.GIC700.ITS0.pvbus_m[0].pvbusmaster	PVBusMaster
GIC700.ITS0.bus_subordinate	PVBusSlave
GIC700.rd_0	GICv4RedistributorInternal
GIC700.rd_0_0	GICv4RedistributorInternal
GIC700.rd_0_0_0	GICv4RedistributorInternal
GIC700.rd_0_0_0_0	GICv3Redistributor
GIC700.rd_0_0_0_0.ExportTest.GIC700.rd_0_0_0_0.pvbus_m[0].pvbusmaster	PVBusMaster
GIC700.rd_t1	GICv3Distributor
GIC700.rd_t1.ExportTest.GIC700.rd_t1.pvbus_m[0].pvbusmaster	PVBusMaster

### Ports for GIC700

Port	Direction	Protocol	Description
axi_stream_msi_s	slave	PVBus	AXI Stream Interface ports for MSI (message-signalled interrupt) translation, in GIC-700. Typically the SMMU's TCU connects to this port for MSI.
chip_id	slave	Value	chip_id port for multichip operation
cpu_active_s	slave	Signal	CPUActive pins.
cpu_wake_request	master	Signal	-
extended_ppi_in_0	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 0.
extended_ppi_in_100	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 100
extended_ppi_in_101	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 101
extended_ppi_in_102	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 102
extended_ppi_in_103	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 103
extended_ppi_in_104	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 104
extended_ppi_in_105	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 105
extended_ppi_in_106	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 106
extended_ppi_in_107	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 107
extended_ppi_in_108	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 108
extended_ppi_in_109	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 109
extended_ppi_in_10	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 10.
extended_ppi_in_110	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 110
extended_ppi_in_111	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 111
extended_ppi_in_112	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 112
extended_ppi_in_113	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 113
extended_ppi_in_114	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 114
extended_ppi_in_115	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 115
extended_ppi_in_116	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 116
extended_ppi_in_117	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 117
extended_ppi_in_118	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 118

Port	Direction	Protocol	Description
extended_ppi_in_119	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 119
extended_ppi_in_11	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 11.
extended_ppi_in_120	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 120
extended_ppi_in_121	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 121
extended_ppi_in_122	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 122
extended_ppi_in_123	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 123
extended_ppi_in_124	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 124
extended_ppi_in_125	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 125
extended_ppi_in_126	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 126
extended_ppi_in_127	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 127
extended_ppi_in_128	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 128
extended_ppi_in_129	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 129
extended_ppi_in_12	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 12.
extended_ppi_in_130	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 130
extended_ppi_in_131	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 131
extended_ppi_in_132	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 132
extended_ppi_in_133	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 133
extended_ppi_in_134	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 134
extended_ppi_in_135	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 135
extended_ppi_in_136	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 136
extended_ppi_in_137	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 137
extended_ppi_in_138	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 138
extended_ppi_in_139	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 139
extended_ppi_in_13	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 13.
extended_ppi_in_140	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 140
extended_ppi_in_141	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 141
extended_ppi_in_142	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 142
extended_ppi_in_143	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 143
extended_ppi_in_144	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 144
extended_ppi_in_145	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 145
extended_ppi_in_146	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 146
extended_ppi_in_147	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 147
extended_ppi_in_148	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 148
extended_ppi_in_149	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 149
extended_ppi_in_14	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 14.
extended_ppi_in_150	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 150
extended_ppi_in_151	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 151
extended_ppi_in_152	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 152
extended_ppi_in_153	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 153
extended_ppi_in_154	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 154

Port	Direction	Protocol	Description
extended_ppi_in_155	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 155
extended_ppi_in_156	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 156
extended_ppi_in_157	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 157
extended_ppi_in_158	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 158
extended_ppi_in_159	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 159
extended_ppi_in_15	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 15.
extended_ppi_in_160	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 160
extended_ppi_in_161	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 161
extended_ppi_in_162	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 162
extended_ppi_in_163	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 163
extended_ppi_in_164	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 164
extended_ppi_in_165	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 165
extended_ppi_in_166	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 166
extended_ppi_in_167	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 167
extended_ppi_in_168	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 168
extended_ppi_in_169	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 169
extended_ppi_in_16	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 16.
extended_ppi_in_170	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 170
extended_ppi_in_171	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 171
extended_ppi_in_172	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 172
extended_ppi_in_173	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 173
extended_ppi_in_174	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 174
extended_ppi_in_175	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 175
extended_ppi_in_176	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 176
extended_ppi_in_177	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 177
extended_ppi_in_178	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 178
extended_ppi_in_179	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 179
extended_ppi_in_17	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 17.
extended_ppi_in_180	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 180
extended_ppi_in_181	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 181
extended_ppi_in_182	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 182
extended_ppi_in_183	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 183
extended_ppi_in_184	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 184
extended_ppi_in_185	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 185
extended_ppi_in_186	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 186
extended_ppi_in_187	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 187
extended_ppi_in_188	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 188
extended_ppi_in_189	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 189
extended_ppi_in_18	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 18.
extended_ppi_in_190	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 190

Port	Direction	Protocol	Description
extended_ppi_in_191	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 191
extended_ppi_in_192	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 192
extended_ppi_in_193	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 193
extended_ppi_in_194	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 194
extended_ppi_in_195	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 195
extended_ppi_in_196	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 196
extended_ppi_in_197	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 197
extended_ppi_in_198	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 198
extended_ppi_in_199	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 199
extended_ppi_in_19	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 19.
extended_ppi_in_1	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 1.
extended_ppi_in_200	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 200
extended_ppi_in_201	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 201
extended_ppi_in_202	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 202
extended_ppi_in_203	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 203
extended_ppi_in_204	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 204
extended_ppi_in_205	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 205
extended_ppi_in_206	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 206
extended_ppi_in_207	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 207
extended_ppi_in_208	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 208
extended_ppi_in_209	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 209
extended_ppi_in_20	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 20.
extended_ppi_in_210	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 210
extended_ppi_in_211	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 211
extended_ppi_in_212	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 212
extended_ppi_in_213	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 213
extended_ppi_in_214	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 214
extended_ppi_in_215	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 215
extended_ppi_in_216	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 216
extended_ppi_in_217	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 217
extended_ppi_in_218	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 218
extended_ppi_in_219	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 219
extended_ppi_in_21	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 21.
extended_ppi_in_220	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 220
extended_ppi_in_221	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 221
extended_ppi_in_222	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 222
extended_ppi_in_223	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 223
extended_ppi_in_224	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 224
extended_ppi_in_225	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 225
extended_ppi_in_226	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 226

Port	Direction	Protocol	Description
extended_ppi_in_227	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 227
extended_ppi_in_228	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 228
extended_ppi_in_229	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 229
extended_ppi_in_22	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 22.
extended_ppi_in_230	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 230
extended_ppi_in_231	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 231
extended_ppi_in_232	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 232
extended_ppi_in_233	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 233
extended_ppi_in_234	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 234
extended_ppi_in_235	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 235
extended_ppi_in_236	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 236
extended_ppi_in_237	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 237
extended_ppi_in_238	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 238
extended_ppi_in_239	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 239
extended_ppi_in_23	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 23.
extended_ppi_in_240	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 240
extended_ppi_in_241	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 241
extended_ppi_in_242	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 242
extended_ppi_in_243	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 243
extended_ppi_in_244	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 244
extended_ppi_in_245	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 245
extended_ppi_in_246	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 246
extended_ppi_in_247	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 247
extended_ppi_in_248	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 248
extended_ppi_in_249	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 249
extended_ppi_in_24	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 24.
extended_ppi_in_250	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 250
extended_ppi_in_251	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 251
extended_ppi_in_252	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 252
extended_ppi_in_253	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 253
extended_ppi_in_254	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 254
extended_ppi_in_255	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 255
extended_ppi_in_25	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 25.
extended_ppi_in_26	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 26.
extended_ppi_in_27	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 27.
extended_ppi_in_28	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 28.
extended_ppi_in_29	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 29.
extended_ppi_in_2	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 2.
extended_ppi_in_30	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 30.
extended_ppi_in_31	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 31.



Port	Direction	Protocol	Description
extended_ppi_in_32	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 32.
extended_ppi_in_33	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 33.
extended_ppi_in_34	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 34.
extended_ppi_in_35	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 35.
extended_ppi_in_36	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 36.
extended_ppi_in_37	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 37.
extended_ppi_in_38	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 38.
extended_ppi_in_39	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 39.
extended_ppi_in_3	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 3.
extended_ppi_in_40	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 40.
extended_ppi_in_41	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 41.
extended_ppi_in_42	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 42.
extended_ppi_in_43	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 43.
extended_ppi_in_44	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 44.
extended_ppi_in_45	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 45.
extended_ppi_in_46	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 46.
extended_ppi_in_47	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 47.
extended_ppi_in_48	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 48.
extended_ppi_in_49	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 49.
extended_ppi_in_4	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 4.
extended_ppi_in_50	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 50.
extended_ppi_in_51	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 51.
extended_ppi_in_52	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 52.
extended_ppi_in_53	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 53.
extended_ppi_in_54	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 54.
extended_ppi_in_55	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 55.
extended_ppi_in_56	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 56.
extended_ppi_in_57	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 57.
extended_ppi_in_58	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 58.
extended_ppi_in_59	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 59.
extended_ppi_in_5	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 5.
extended_ppi_in_60	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 60.
extended_ppi_in_61	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 61.
extended_ppi_in_62	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 62.
extended_ppi_in_63	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 63.
extended_ppi_in_64	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 64.
extended_ppi_in_65	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 65.
extended_ppi_in_66	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 66.
extended_ppi_in_67	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 67.
extended_ppi_in_68	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 68.



Port	Direction	Protocol	Description
extended_ppi_in_69	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 69
extended_ppi_in_6	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 6.
extended_ppi_in_70	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 70
extended_ppi_in_71	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 71
extended_ppi_in_72	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 72
extended_ppi_in_73	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 73
extended_ppi_in_74	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 74
extended_ppi_in_75	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 75
extended_ppi_in_76	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 76
extended_ppi_in_77	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 77
extended_ppi_in_78	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 78
extended_ppi_in_79	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 79
extended_ppi_in_7	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 7.
extended_ppi_in_80	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 80
extended_ppi_in_81	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 81
extended_ppi_in_82	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 82
extended_ppi_in_83	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 83
extended_ppi_in_84	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 84
extended_ppi_in_85	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 85
extended_ppi_in_86	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 86
extended_ppi_in_87	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 87
extended_ppi_in_88	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 88
extended_ppi_in_89	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 89
extended_ppi_in_8	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 8.
extended_ppi_in_90	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 90
extended_ppi_in_91	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 91
extended_ppi_in_92	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 92
extended_ppi_in_93	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 93
extended_ppi_in_94	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 94
extended_ppi_in_95	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 95
extended_ppi_in_96	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 96
extended_ppi_in_97	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 97
extended_ppi_in_98	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 98
extended_ppi_in_99	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 99
extended_ppi_in_9	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 9.
extended_spi_in	slave	Signal	Extended Shared peripheral interrupts.
icdrdt_out	master	PVBus	AXI Stream Interface. This is used only when local cross chip addressing is enabled at the moment
icrdt_in	slave	PVBus	-
po_reset	slave	Signal	Resets.
ppi_in_0	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 0.

Port	Direction	Protocol	Description
ppi_in_100	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 100
ppi_in_101	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 101
ppi_in_102	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 102
ppi_in_103	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 103
ppi_in_104	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 104
ppi_in_105	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 105
ppi_in_106	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 106
ppi_in_107	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 107
ppi_in_108	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 108
ppi_in_109	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 109
ppi_in_10	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 10.
ppi_in_110	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 110
ppi_in_111	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 111
ppi_in_112	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 112
ppi_in_113	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 113
ppi_in_114	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 114
ppi_in_115	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 115
ppi_in_116	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 116
ppi_in_117	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 117
ppi_in_118	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 118
ppi_in_119	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 119
ppi_in_11	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 11.
ppi_in_120	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 120
ppi_in_121	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 121
ppi_in_122	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 122
ppi_in_123	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 123
ppi_in_124	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 124
ppi_in_125	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 125
ppi_in_126	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 126
ppi_in_127	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 127
ppi_in_128	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 128
ppi_in_129	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 129
ppi_in_12	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 12.
ppi_in_130	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 130
ppi_in_131	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 131
ppi_in_132	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 132
ppi_in_133	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 133
ppi_in_134	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 134
ppi_in_135	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 135
ppi_in_136	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 136

Port	Direction	Protocol	Description
ppi_in_137	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 137
ppi_in_138	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 138
ppi_in_139	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 139
ppi_in_13	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 13.
ppi_in_140	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 140
ppi_in_141	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 141
ppi_in_142	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 142
ppi_in_143	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 143
ppi_in_144	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 144
ppi_in_145	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 145
ppi_in_146	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 146
ppi_in_147	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 147
ppi_in_148	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 148
ppi_in_149	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 149
ppi_in_14	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 14.
ppi_in_150	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 150
ppi_in_151	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 151
ppi_in_152	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 152
ppi_in_153	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 153
ppi_in_154	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 154
ppi_in_155	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 155
ppi_in_156	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 156
ppi_in_157	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 157
ppi_in_158	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 158
ppi_in_159	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 159
ppi_in_15	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 15.
ppi_in_160	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 160
ppi_in_161	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 161
ppi_in_162	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 162
ppi_in_163	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 163
ppi_in_164	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 164
ppi_in_165	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 165
ppi_in_166	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 166
ppi_in_167	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 167
ppi_in_168	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 168
ppi_in_169	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 169
ppi_in_16	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 16.
ppi_in_170	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 170
ppi_in_171	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 171
ppi_in_172	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 172

Port	Direction	Protocol	Description
ppi_in_173	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 173
ppi_in_174	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 174
ppi_in_175	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 175
ppi_in_176	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 176
ppi_in_177	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 177
ppi_in_178	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 178
ppi_in_179	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 179
ppi_in_17	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 17.
ppi_in_180	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 180
ppi_in_181	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 181
ppi_in_182	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 182
ppi_in_183	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 183
ppi_in_184	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 184
ppi_in_185	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 185
ppi_in_186	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 186
ppi_in_187	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 187
ppi_in_188	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 188
ppi_in_189	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 189
ppi_in_18	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 18.
ppi_in_190	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 190
ppi_in_191	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 191
ppi_in_192	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 192
ppi_in_193	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 193
ppi_in_194	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 194
ppi_in_195	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 195
ppi_in_196	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 196
ppi_in_197	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 197
ppi_in_198	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 198
ppi_in_199	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 199
ppi_in_19	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 19.
ppi_in_1	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 1.
ppi_in_200	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 200
ppi_in_201	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 201
ppi_in_202	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 202
ppi_in_203	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 203
ppi_in_204	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 204
ppi_in_205	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 205
ppi_in_206	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 206
ppi_in_207	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 207
ppi_in_208	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 208

Port	Direction	Protocol	Description
ppi_in_209	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 209
ppi_in_20	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 20.
ppi_in_210	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 210
ppi_in_211	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 211
ppi_in_212	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 212
ppi_in_213	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 213
ppi_in_214	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 214
ppi_in_215	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 215
ppi_in_216	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 216
ppi_in_217	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 217
ppi_in_218	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 218
ppi_in_219	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 219
ppi_in_21	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 21.
ppi_in_220	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 220
ppi_in_221	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 221
ppi_in_222	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 222
ppi_in_223	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 223
ppi_in_224	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 224
ppi_in_225	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 225
ppi_in_226	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 226
ppi_in_227	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 227
ppi_in_228	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 228
ppi_in_229	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 229
ppi_in_22	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 22.
ppi_in_230	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 230
ppi_in_231	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 231
ppi_in_232	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 232
ppi_in_233	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 233
ppi_in_234	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 234
ppi_in_235	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 235
ppi_in_236	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 236
ppi_in_237	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 237
ppi_in_238	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 238
ppi_in_239	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 239
ppi_in_23	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 23.
ppi_in_240	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 240
ppi_in_241	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 241
ppi_in_242	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 242
ppi_in_243	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 243
ppi_in_244	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 244

Port	Direction	Protocol	Description
ppi_in_245	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 245
ppi_in_246	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 246
ppi_in_247	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 247
ppi_in_248	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 248
ppi_in_249	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 249
ppi_in_24	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 24.
ppi_in_250	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 250
ppi_in_251	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 251
ppi_in_252	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 252
ppi_in_253	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 253
ppi_in_254	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 254
ppi_in_255	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 255
ppi_in_25	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 25.
ppi_in_26	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 26.
ppi_in_27	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 27.
ppi_in_28	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 28.
ppi_in_29	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 29.
ppi_in_2	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 2.
ppi_in_30	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 30.
ppi_in_31	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 31.
ppi_in_32	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 32.
ppi_in_33	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 33.
ppi_in_34	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 34.
ppi_in_35	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 35.
ppi_in_36	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 36.
ppi_in_37	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 37.
ppi_in_38	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 38.
ppi_in_39	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 39.
ppi_in_3	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 3.
ppi_in_40	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 40.
ppi_in_41	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 41.
ppi_in_42	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 42.
ppi_in_43	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 43.
ppi_in_44	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 44.
ppi_in_45	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 45.
ppi_in_46	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 46.
ppi_in_47	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 47.
ppi_in_48	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 48.
ppi_in_49	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 49.
ppi_in_4	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 4.

Port	Direction	Protocol	Description
ppi_in_50	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 50.
ppi_in_51	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 51.
ppi_in_52	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 52.
ppi_in_53	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 53.
ppi_in_54	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 54.
ppi_in_55	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 55.
ppi_in_56	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 56.
ppi_in_57	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 57.
ppi_in_58	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 58.
ppi_in_59	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 59.
ppi_in_5	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 5.
ppi_in_60	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 60.
ppi_in_61	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 61.
ppi_in_62	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 62.
ppi_in_63	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 63.
ppi_in_64	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 64.
ppi_in_65	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 65.
ppi_in_66	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 66.
ppi_in_67	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 67.
ppi_in_68	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 68.
ppi_in_69	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 69.
ppi_in_6	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 6.
ppi_in_70	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 70.
ppi_in_71	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 71.
ppi_in_72	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 72.
ppi_in_73	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 73.
ppi_in_74	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 74.
ppi_in_75	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 75.
ppi_in_76	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 76.
ppi_in_77	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 77.
ppi_in_78	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 78.
ppi_in_79	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 79.
ppi_in_7	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 7.
ppi_in_80	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 80.
ppi_in_81	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 81.
ppi_in_82	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 82.
ppi_in_83	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 83.
ppi_in_84	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 84.
ppi_in_85	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 85.
ppi_in_86	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 86.



Port	Direction	Protocol	Description
ppi_in_87	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 87
ppi_in_88	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 88
ppi_in_89	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 89
ppi_in_8	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 8.
ppi_in_90	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 90
ppi_in_91	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 91
ppi_in_92	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 92
ppi_in_93	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 93
ppi_in_94	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 94
ppi_in_95	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 95
ppi_in_96	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 96
ppi_in_97	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 97
ppi_in_98	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 98
ppi_in_99	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 99
ppi_in_9	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 9.
pvbus_m	master	PVBus	Memory bus out: transactions generated by the IRI.
pvbus_s	slave	PVBus	Memory bus in: memory-mapped register accesses are accepted on this interface.
redistributor_m	master	GICv3Comms	Input from and output to CPU interface.
reset	slave	Signal	Resets.
spi_in	slave	Signal	Shared peripheral interrupts.
wake_request	master	Signal	Power management outputs.

## Parameters for GIC700

### CPU-affinities

A comma separated list of dotted quads containing the affinities of all PEs connected to this IRI. If CPU-affinities-file is specified, this parameter is ignored.

Type: `string`

Default value: `N/A`

### CPU-affinities-file

A file containing comma separated list of dotted quads containing the affinities of all PEs connected to this IRI. If this parameter is specified, CPU-affinities parameter will be ignored even when it is given.

Type: `string`

Default value: `N/A`



**GICD\_CTLR-DS-1-means-secure-only**

If GICD\_CTLR.DS=1, GICD supports a single security state which is secure if this is true, otherwise is non-secure.

Type: `bool`

Default value: `false`

**GICD\_TYPER2**

GICD\_TYPER2 value containing VID and VIL to define the width of vPEID for GICv4.1.

Type: `uint32_t`

Default value: `0`

**GICR-clear-enable-supported**

When true, this sets the value of the RO bit GICR\_CTLR.CES with the value of the parameter `allow-LPIEN-clear`, making it visible to software.

Type: `bool`

Default value: `false`

**GICR-invalidate-registers-implemented**

When true, the registers GICR\_INVLPIR, GICR\_INVALLR and GICR\_SYNCRR are implemented.

Type: `bool`

Default value: `false`

**IIDR**

GICD\_IIDR, GICR\_IIDR and GITS\_IIDR value. Defaults to the latest modelled revision.

Type: `uint32_t`

Default value: `0x0404043B`

**ITS-ID-bits**

Number of interrupt bits supported by ITS.

Type: `uint8_t`

Default value: `16`

**ITS-collection-ID-bits**

Number of collection bits supported by ITS (optional parameter, 0 => 16bits support and GITS\_TYPER.CIL=0).

Type: `uint8_t`

Default value: 8

### **ITS-count**

Number of Interrupt Translation Services to be instantiated (0=none).

Type: `uint8_t`

Default value: 1

### **ITS-device-bits**

Number of bits supported for ITS device IDs.

Type: `uint8_t`

Default value: 16

### **ITS-enable-itt-address-verification**

If true, a transaction will be sent to ITT Address for verification.

Type: `bool`

Default value: `false`

### **ITS-hardware-collection-count**

Number of hardware collections held exclusively in the ITS.

Type: `uint8_t`

Default value: 0

### **ITS-shared-vPE-table**

Number of affinity levels to which the vPE configuration table is shared. This parameter is valid when `has-gicv4.1` is true.

Type: `uint8_t`

Default value: 0

### **ITS-vmovp-bit**

Device supports software issuing a VMOVP to only one of the ITSs that has a mapping for a vPE. The device itself ensures synchronization of the VMOVP command across all ITSs that have mapping for that vPE.

Type: `bool`

Default value: `false`

**PPI-count**

Selects the number of PPI available for each PE

**8**

id22-27,29,30

**12**

id 20-31

**16**

id 16-31.

Type: `int`

Default value: 16

**SPI-blocks**

Number of SPI blocks supported by the IRI, each block contains 32 SPIs.

Type: `int`

Default value: 62

**add-output-cpu-wake-request-signal-from-redistributor**

If true, the redistributor will have the output signal `cpu_wake_request` from GIC to DSU and if false, the signals are not added to the redistributor.

Type: `bool`

Default value: `false`

**affinity-width**

A dotted quad indicating the bitwidth of fields at each affinity level.

Type: `string`

Default value: "4.8.8.8"

**allow-LPIEN-clear**

Allow RW behaviour on GICR\_CTLR.LPIEN instead of set once.

Type: `bool`

Default value: `true`

**chip-count**

The total number of chips supported.

Type: `int`

Default value: 16

**chip-id**

Chip ID for multichip operation.

Type: `uint8_t`

Default value: 0

**chip-select-affinity-level**

Affinity level 2 or 3 can be used for chip select.

Type: `int`

Default value: 3

**clear-ISPENDR-bit-for-level-sensitive-interrupt-when-acknowledged**

If true, acknowledgement of a level sensitive interrupt clears the corresponding bit in the ISPENDR register.

Type: `bool`

Default value: `false`

**common-lpi-configuration**

Describes which re-distributors share (and must be configured with the same) LPI configuration table as described in GICR\_TYPER( 0:All, 1:A.x.x.x, 2:A.B.x.x, 3:A.B.C.x).

Type: `int`

Default value: 0

**common-vPE-table-affinity**

Affinity value list in the form of 'a.b.c.d, e.f.g.h, etc' given to the ITS(s), where a.b.c.d corresponds to ITS0, e.f.g.h corresponds to ITS1 and so on. Under an affinity value the vPE configuration table is shared among redistributors where the level to be shared is defined by ITS-shared-vPE-table. This parameter is valid when `has-gicv4.1` is true.

Type: `string`

Default value: N/A

**consolidators**

Specify consolidators' base addresses, interrupt line counts and base interrupt IDs, in the form:

```
'baseAddr0:itlineCount0:baseINTID0, baseAddr1:itlineCount1:baseINTID1'
```

For example:

```
'0x3f100000:64:4096, 0x3f200000:64:4224'
```

The consolidators' count is inferred from the list (maximum of 4). If not specified, the component contains no consolidators.

Type: `string`

Default value: `""`

### **cross-chip-AMBA-is-ACE**

Indicates the AMBA protocol that the cross-chip interface uses. If true, the cross-chip interface uses the ACE5-Lite protocol. Otherwise, it uses the AXI5-Stream protocol.

Type: `bool`

Default value: `false`

### **enable-local-cross-chip-addressing**

If true, each distributor in a multichip system will use routing table address (ADDR) values programmed by local software, and will not be overwritten by the default chip. Otherwise, all the distributors will share the same routing table address values programmed in the default chip.

Type: `bool`

Default value: `false`

### **enable-multiple-views-feature**

If true, multiple view feature will provide multiple programming views which can be used by multiple hypervisors.

Type: `bool`

Default value: `false`

### **enabled**

Enable GICv3 functionality; when false the component is inactive.

Type: `bool`

Default value: `true`

### **extended-ppi-count**

Number of extended PPI supported.

Type: `unsigned`

Default value: 32

**gicp-allow-ns-reset**

If true, non-secure read/write access to GICP register is allowed at reset. Not allowed otherwise. This emulates gicp\_allow\_ns tie-off signal.

Type: bool

Default value: true

**gict-allow-ns-reset**

If true, non-secure read/write access to GICT register is allowed at reset. Not allowed otherwise. This emulates gict\_allow\_ns tie-off signal.

Type: bool

Default value: true

**has-gicv4.1**

Enable GICv4.1 functionality; when false the component is inactive.

Type: bool

Default value: true

**has-two-security-states**

If true, has two security states.

Type: bool

Default value: true

**has\_VPENDBASER-dirty-flag-on-load**

GICR\_VPENDBASER.Dirty reflects transient loading state when valid=1.

Type: bool

Default value: false

**has\_nmi**

Enable support for Non-maskable Interrupts (NMIs).

Type: bool

Default value: false

**max-cores-supported-by-GCI**

GCI can support 1-64 cores. Maximum supported cores in GCI is configurable by device.

Type: `int`

Default value: `64`

**max-pe-on-chip**

Maximum number of cores on any single chip. This will be used to identify the target chip and core when multichip operation is enabled.

Type: `int`

Default value: `4`

**multichip-threaded-dgi**

Enable sending of multichip DGI messages in a separate thread, when multichip operation is enabled.

Type: `bool`

Default value: `true`

**output\_attributes**

User-defined transform to be applied to bus attributes like ManagerID, ExtendedID or UserFlags. Currently, only works for MPAM Attributes encoding into bus attributes.

Type: `string`

Default value: `"ExtendedID[62:55]=MPAM_PMG, ExtendedID[54:39]=MPAM_PARTID, ExtendedID[38]=MPAM_NS"`

**print-memory-map**

Print memory map to stdout.

Type: `bool`

Default value: `false`

**prog-mpidr**

Whether software or hardware can remove cores from a GIC configuration.

**0**

none

**1**

prog - Secure software to remove cores during the boot up of a system.

## 2

strap - enables hardware to remove cores as GIC exits reset.

Type: `unsigned`

Default value: 0

### **redistributor-group**

Redistributor grouping information with affinity as JSON :

```
{
  "0": [
    "0.0.0.0",
    "0.0.0.1"
  ],
  "1": [
    "0.0.1.0",
    "0.0.1.1"
  ]
}
```

where RD with quad 0.0.0.0 and 0.0.0.1 belongs to RD group 0. All the RDs belong to one group when this parameter is not given.

Type: `string`

Default value: N/A

### **redistributor-group-file**

File path to redistributor grouping information with affinity as JSON.

The file uses the same format as `redistributor-group` parameter.

Type: `string`

Default value: N/A

### **redistributor-power-managed-by-pwrr**

GIC-700 redistributor power management is done by updating GICR\_PWRR register.

Type: `bool`

Default value: `true`

### **reg-base**

GIC-700 base address for memory-mapped register.

Type: `uint64_t`

Default value: `0x2c010000`



**reg-base-per-redistributor**

Base address for each redistributor in the form:

```
0.0.0.0=0x2c010000, 0.0.0.1=0x2c020000
```

All redistributors must be specified and this overrides the `reg-base` parameter (except that `reg-base` will still be used for the top-level redistributor). If `reg-base-per-redistributor-file` is specified, this parameter is ignored.

Type: `string`

Default value: `""`

**view-id-bits-offset**

The offset of the view-id bits in the address. Effective only when multiple views feature is enabled. Accepted values will depend on the configuration affecting the GIC memory footprint, such as the input to the parameter 'CPU-affinities'. For example, for a 32MB-sized view, the value of this parameter should be 25.

Type: `int`

Default value: 0

## 3.195 GIC700\_Filter

Defined in `LISA/GIC700.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

**About GIC700\_Filter**

GIC700 and GIC700\_Filter are minimal models of an Arm&reg GIC-700 Generic Interrupt Controller, suitable for single-chip or multichip systems. They support the following multichip operations:

- Multichip configuration through distributor registers.
- SPI.
- SGI.
- Physical LPI.
- Physical LPI command.

- Virtual LPI.
- Virtual LPI command.

The models provide a simple configuration interface that allows designers to introduce GIC700-like functionality to systems, while only implementing the architectural behavior, as defined by the GICv3/GICv4 architecture.

All implementation-specific registers and functionality are implemented.

As with the other GIC components, there are two variants of the model with slightly different memory interfaces. Both GIC700 and GIC700\_Filter have a `pvbuss_s` port for register accesses and a `pvbuss_m` port for the LPI-related traffic from redistributors and the ITS.

In addition, the GIC700\_Filter variant has a `pvbuss_filtermiss_m` port, to which any transactions on the `pvbuss_s` port and not directed to a 4 KB page used by the GIC are forwarded unmodified. Such transactions are terminated in the component when using the GIC700 variant.

GIC700 and GIC700\_Filter support an extended interrupt range, with 64 more PPIs and 1024 more SPIs. To program additional interrupts, refer to the GICv3/GICv4 specification.

GIC700 and GIC700\_Filter support direct injection of virtual interrupts. This reduces the overhead of trapping to hypervisor for the generation of virtual interrupts. In the GICv3/GICv4 specification, direct injection is supported in two different versions, GICv4.0 and GICv4.1. GIC700 and GIC700\_Filter support GICv4.1.

It is recommended to use GIC700 instead of GIC700\_Filter in most cases.

The GIC700 model supports multiple revisions of the IP. By default, its behavior is set to the latest revision supported. Parameters can be used to enable or disable features that are specific to earlier revisions. The LCA (Local Cross-chip Addressing) feature which is added in r2p0 is disabled by default and is enabled by setting the parameter `enable-local-cross-chip-addressing` to true. The model's registers follow the specification for the latest IP revision supported.



Note

- Set the FASTSIM\_GIC\_MEMORY\_MAP environment variable to 1 to print to stderr the memory map of any GICv3 or later models that are included in the platform being run.
- For writes to `GITS_TRANSLATE64R`, the MSIRewriter component can be used to create the correct data transactions based on the device ID. For more details about `GITS_TRANSLATE64R`, see [MSIRewriter](#).

## Limitations

- The following multichip operations are not yet supported:
  - vPE control.
  - Power control.
- There is no support for the VMOV command.

- Limited testing has been performed for multichip LPI operations, so the model might contain defects.
- One ITS is supported for each GIC for multichip operations.

### ManagerID, ExtendedID, and UserFlags

This table shows how this model encodes information in the `ManagerID`, `ExtendedID`, and `UserFlags` bus attributes:

**Table 3-682: ManagerID, ExtendedID, and UserFlags**

PVBus attribute	Bits used	Property encoded	Notes
ManagerID	63:0	Manager ID that invoked the register access	–
ExtendedID	–	–	Not used
UserFlags	–	–	Not used



Note

The `pvbus_m` port does not use `ManagerID`, `ExtendedID`, or `UserFlags`.

### Iris and MTI instances for GIC700\_Filter

This model has the following Iris instances:

Name	Instance type
GIC700_Filter	GIC700
GIC700_Filter.GICV3_ProtocolChecker	GICv3ProtocolChecker
GIC700_Filter.ITS0	GICv4InterruptTranslationService
GIC700_Filter.ITS0.ExportTest.GIC700_Filter.ITS0.pvbus_m[0].pvbusmaster	PVBusMaster
GIC700_Filter.ITS0.bus_subordinate	PVBusSlave
GIC700_Filter.rd_0	GICv4RedistributorInternal
GIC700_Filter.rd_0_0	GICv4RedistributorInternal
GIC700_Filter.rd_0_0_0	GICv4RedistributorInternal
GIC700_Filter.rd_0_0_0_0	GICv3Redistributor
GIC700_Filter.rd_0_0_0_0.ExportTest.GIC700_Filter.rd_0_0_0_0.pvbus_m[0].pvbusmaster	PVBusMaster
GIC700_Filter.rd_tl	GICv3Distributor
GIC700_Filter.rd_tl.ExportTest.GIC700_Filter.rd_tl.pvbus_m[0].pvbusmaster	PVBusMaster

This model has the following MTI trace components:

Name	Component type
GIC700_Filter.GICV3_ProtocolChecker	GICv3ProtocolChecker
GIC700_Filter.ITS0	GICv4InterruptTranslationService
GIC700_Filter.ITS0.ExportTest.GIC700_Filter.ITS0.pvbus_m[0].pvbusmaster	PVBusMaster

Name	Component type
GIC700_Filter.ITS0.bus_subordinate	PVBusSlave
GIC700_Filter.rd_0	GICv4RedistributorInternal
GIC700_Filter.rd_0_0	GICv4RedistributorInternal
GIC700_Filter.rd_0_0_0	GICv4RedistributorInternal
GIC700_Filter.rd_0_0_0_0	GICv3Redistributor
GIC700_Filter.rd_0_0_0_0.ExportTest.GIC700_Filter.rd_0_0_0_0.pvbus_m[0].pvbusmaster	PVBusMaster
GIC700_Filter.rd_tl	GICv3Distributor
GIC700_Filter.rd_tl.ExportTest.GIC700_Filter.rd_tl.pvbus_m[0].pvbusmaster	PVBusMaster

### Ports for GIC700\_Filter

Port	Direction	Protocol	Description
axi_stream_msi_s	slave	PVBus	AXI Stream Interface ports for MSI (message-signalled interrupt) translation, in GIC-700. Typically the SMMU's TCU connects to this port for MSI.
chip_id	slave	Value	chip_id port for multichip operation
cpu_active_s	slave	Signal	CPUActive pins.
extended_ppi_in_0	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 0.
extended_ppi_in_100	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 100
extended_ppi_in_101	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 101
extended_ppi_in_102	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 102
extended_ppi_in_103	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 103
extended_ppi_in_104	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 104
extended_ppi_in_105	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 105
extended_ppi_in_106	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 106
extended_ppi_in_107	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 107
extended_ppi_in_108	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 108
extended_ppi_in_109	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 109
extended_ppi_in_10	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 10.
extended_ppi_in_110	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 110
extended_ppi_in_111	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 111
extended_ppi_in_112	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 112
extended_ppi_in_113	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 113
extended_ppi_in_114	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 114
extended_ppi_in_115	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 115
extended_ppi_in_116	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 116
extended_ppi_in_117	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 117
extended_ppi_in_118	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 118
extended_ppi_in_119	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 119
extended_ppi_in_11	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 11.
extended_ppi_in_120	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 120
extended_ppi_in_121	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 121

Port	Direction	Protocol	Description
extended_ppi_in_122	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 122
extended_ppi_in_123	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 123
extended_ppi_in_124	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 124
extended_ppi_in_125	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 125
extended_ppi_in_126	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 126
extended_ppi_in_127	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 127
extended_ppi_in_128	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 128
extended_ppi_in_129	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 129
extended_ppi_in_12	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 12.
extended_ppi_in_130	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 130
extended_ppi_in_131	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 131
extended_ppi_in_132	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 132
extended_ppi_in_133	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 133
extended_ppi_in_134	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 134
extended_ppi_in_135	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 135
extended_ppi_in_136	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 136
extended_ppi_in_137	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 137
extended_ppi_in_138	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 138
extended_ppi_in_139	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 139
extended_ppi_in_13	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 13.
extended_ppi_in_140	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 140
extended_ppi_in_141	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 141
extended_ppi_in_142	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 142
extended_ppi_in_143	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 143
extended_ppi_in_144	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 144
extended_ppi_in_145	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 145
extended_ppi_in_146	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 146
extended_ppi_in_147	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 147
extended_ppi_in_148	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 148
extended_ppi_in_149	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 149
extended_ppi_in_14	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 14.
extended_ppi_in_150	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 150
extended_ppi_in_151	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 151
extended_ppi_in_152	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 152
extended_ppi_in_153	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 153
extended_ppi_in_154	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 154
extended_ppi_in_155	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 155
extended_ppi_in_156	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 156
extended_ppi_in_157	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 157
extended_ppi_in_158	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 158

Port	Direction	Protocol	Description
extended_ppi_in_159	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 159
extended_ppi_in_15	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 15.
extended_ppi_in_160	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 160
extended_ppi_in_161	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 161
extended_ppi_in_162	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 162
extended_ppi_in_163	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 163
extended_ppi_in_164	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 164
extended_ppi_in_165	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 165
extended_ppi_in_166	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 166
extended_ppi_in_167	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 167
extended_ppi_in_168	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 168
extended_ppi_in_169	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 169
extended_ppi_in_16	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 16.
extended_ppi_in_170	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 170
extended_ppi_in_171	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 171
extended_ppi_in_172	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 172
extended_ppi_in_173	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 173
extended_ppi_in_174	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 174
extended_ppi_in_175	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 175
extended_ppi_in_176	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 176
extended_ppi_in_177	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 177
extended_ppi_in_178	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 178
extended_ppi_in_179	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 179
extended_ppi_in_17	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 17.
extended_ppi_in_180	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 180
extended_ppi_in_181	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 181
extended_ppi_in_182	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 182
extended_ppi_in_183	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 183
extended_ppi_in_184	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 184
extended_ppi_in_185	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 185
extended_ppi_in_186	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 186
extended_ppi_in_187	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 187
extended_ppi_in_188	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 188
extended_ppi_in_189	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 189
extended_ppi_in_18	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 18.
extended_ppi_in_190	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 190
extended_ppi_in_191	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 191
extended_ppi_in_192	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 192
extended_ppi_in_193	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 193
extended_ppi_in_194	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 194

Port	Direction	Protocol	Description
extended_ppi_in_195	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 195
extended_ppi_in_196	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 196
extended_ppi_in_197	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 197
extended_ppi_in_198	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 198
extended_ppi_in_199	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 199
extended_ppi_in_19	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 19.
extended_ppi_in_1	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 1.
extended_ppi_in_200	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 200
extended_ppi_in_201	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 201
extended_ppi_in_202	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 202
extended_ppi_in_203	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 203
extended_ppi_in_204	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 204
extended_ppi_in_205	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 205
extended_ppi_in_206	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 206
extended_ppi_in_207	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 207
extended_ppi_in_208	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 208
extended_ppi_in_209	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 209
extended_ppi_in_20	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 20.
extended_ppi_in_210	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 210
extended_ppi_in_211	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 211
extended_ppi_in_212	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 212
extended_ppi_in_213	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 213
extended_ppi_in_214	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 214
extended_ppi_in_215	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 215
extended_ppi_in_216	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 216
extended_ppi_in_217	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 217
extended_ppi_in_218	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 218
extended_ppi_in_219	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 219
extended_ppi_in_21	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 21.
extended_ppi_in_220	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 220
extended_ppi_in_221	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 221
extended_ppi_in_222	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 222
extended_ppi_in_223	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 223
extended_ppi_in_224	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 224
extended_ppi_in_225	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 225
extended_ppi_in_226	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 226
extended_ppi_in_227	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 227
extended_ppi_in_228	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 228
extended_ppi_in_229	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 229
extended_ppi_in_22	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 22.



Port	Direction	Protocol	Description
extended_ppi_in_230	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 230
extended_ppi_in_231	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 231
extended_ppi_in_232	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 232
extended_ppi_in_233	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 233
extended_ppi_in_234	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 234
extended_ppi_in_235	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 235
extended_ppi_in_236	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 236
extended_ppi_in_237	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 237
extended_ppi_in_238	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 238
extended_ppi_in_239	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 239
extended_ppi_in_23	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 23.
extended_ppi_in_240	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 240
extended_ppi_in_241	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 241
extended_ppi_in_242	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 242
extended_ppi_in_243	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 243
extended_ppi_in_244	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 244
extended_ppi_in_245	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 245
extended_ppi_in_246	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 246
extended_ppi_in_247	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 247
extended_ppi_in_248	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 248
extended_ppi_in_249	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 249
extended_ppi_in_24	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 24.
extended_ppi_in_250	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 250
extended_ppi_in_251	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 251
extended_ppi_in_252	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 252
extended_ppi_in_253	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 253
extended_ppi_in_254	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 254
extended_ppi_in_255	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 255
extended_ppi_in_25	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 25.
extended_ppi_in_26	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 26.
extended_ppi_in_27	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 27.
extended_ppi_in_28	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 28.
extended_ppi_in_29	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 29.
extended_ppi_in_2	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 2.
extended_ppi_in_30	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 30.
extended_ppi_in_31	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 31.
extended_ppi_in_32	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 32.
extended_ppi_in_33	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 33.
extended_ppi_in_34	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 34.
extended_ppi_in_35	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 35.



Port	Direction	Protocol	Description
extended_ppi_in_36	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 36.
extended_ppi_in_37	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 37.
extended_ppi_in_38	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 38.
extended_ppi_in_39	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 39.
extended_ppi_in_3	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 3.
extended_ppi_in_40	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 40.
extended_ppi_in_41	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 41.
extended_ppi_in_42	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 42.
extended_ppi_in_43	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 43.
extended_ppi_in_44	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 44.
extended_ppi_in_45	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 45.
extended_ppi_in_46	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 46.
extended_ppi_in_47	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 47.
extended_ppi_in_48	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 48.
extended_ppi_in_49	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 49.
extended_ppi_in_4	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 4.
extended_ppi_in_50	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 50.
extended_ppi_in_51	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 51.
extended_ppi_in_52	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 52.
extended_ppi_in_53	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 53.
extended_ppi_in_54	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 54.
extended_ppi_in_55	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 55.
extended_ppi_in_56	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 56.
extended_ppi_in_57	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 57.
extended_ppi_in_58	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 58.
extended_ppi_in_59	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 59.
extended_ppi_in_5	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 5.
extended_ppi_in_60	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 60.
extended_ppi_in_61	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 61.
extended_ppi_in_62	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 62.
extended_ppi_in_63	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 63.
extended_ppi_in_64	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 64.
extended_ppi_in_65	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 65.
extended_ppi_in_66	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 66.
extended_ppi_in_67	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 67.
extended_ppi_in_68	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 68.
extended_ppi_in_69	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 69.
extended_ppi_in_6	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 6.
extended_ppi_in_70	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 70.
extended_ppi_in_71	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 71.

Port	Direction	Protocol	Description
extended_ppi_in_72	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 72
extended_ppi_in_73	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 73
extended_ppi_in_74	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 74
extended_ppi_in_75	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 75
extended_ppi_in_76	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 76
extended_ppi_in_77	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 77
extended_ppi_in_78	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 78
extended_ppi_in_79	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 79
extended_ppi_in_7	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 7.
extended_ppi_in_80	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 80
extended_ppi_in_81	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 81
extended_ppi_in_82	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 82
extended_ppi_in_83	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 83
extended_ppi_in_84	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 84
extended_ppi_in_85	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 85
extended_ppi_in_86	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 86
extended_ppi_in_87	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 87
extended_ppi_in_88	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 88
extended_ppi_in_89	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 89
extended_ppi_in_8	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 8.
extended_ppi_in_90	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 90
extended_ppi_in_91	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 91
extended_ppi_in_92	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 92
extended_ppi_in_93	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 93
extended_ppi_in_94	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 94
extended_ppi_in_95	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 95
extended_ppi_in_96	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 96
extended_ppi_in_97	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 97
extended_ppi_in_98	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 98
extended_ppi_in_99	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 99
extended_ppi_in_9	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 9.
extended_spi_in	slave	Signal	Extended Shared peripheral interrupts.
icdrt_out	master	PVBus	AXI Stream Interface. This is used only when local cross chip addressing is enabled at the moment
icrdt_in	slave	PVBus	-
po_reset	slave	Signal	Reset.
ppi_in_0	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 0.
ppi_in_100	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 100
ppi_in_101	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 101
ppi_in_102	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 102
ppi_in_103	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 103

Port	Direction	Protocol	Description
ppi_in_104	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 104
ppi_in_105	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 105
ppi_in_106	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 106
ppi_in_107	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 107
ppi_in_108	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 108
ppi_in_109	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 109
ppi_in_10	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 10.
ppi_in_110	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 110
ppi_in_111	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 111
ppi_in_112	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 112
ppi_in_113	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 113
ppi_in_114	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 114
ppi_in_115	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 115
ppi_in_116	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 116
ppi_in_117	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 117
ppi_in_118	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 118
ppi_in_119	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 119
ppi_in_11	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 11.
ppi_in_120	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 120
ppi_in_121	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 121
ppi_in_122	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 122
ppi_in_123	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 123
ppi_in_124	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 124
ppi_in_125	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 125
ppi_in_126	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 126
ppi_in_127	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 127
ppi_in_128	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 128
ppi_in_129	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 129
ppi_in_12	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 12.
ppi_in_130	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 130
ppi_in_131	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 131
ppi_in_132	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 132
ppi_in_133	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 133
ppi_in_134	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 134
ppi_in_135	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 135
ppi_in_136	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 136
ppi_in_137	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 137
ppi_in_138	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 138
ppi_in_139	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 139
ppi_in_13	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 13.

Port	Direction	Protocol	Description
ppi_in_140	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 140
ppi_in_141	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 141
ppi_in_142	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 142
ppi_in_143	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 143
ppi_in_144	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 144
ppi_in_145	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 145
ppi_in_146	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 146
ppi_in_147	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 147
ppi_in_148	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 148
ppi_in_149	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 149
ppi_in_14	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 14.
ppi_in_150	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 150
ppi_in_151	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 151
ppi_in_152	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 152
ppi_in_153	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 153
ppi_in_154	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 154
ppi_in_155	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 155
ppi_in_156	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 156
ppi_in_157	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 157
ppi_in_158	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 158
ppi_in_159	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 159
ppi_in_15	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 15.
ppi_in_160	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 160
ppi_in_161	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 161
ppi_in_162	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 162
ppi_in_163	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 163
ppi_in_164	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 164
ppi_in_165	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 165
ppi_in_166	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 166
ppi_in_167	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 167
ppi_in_168	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 168
ppi_in_169	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 169
ppi_in_16	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 16.
ppi_in_170	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 170
ppi_in_171	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 171
ppi_in_172	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 172
ppi_in_173	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 173
ppi_in_174	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 174
ppi_in_175	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 175
ppi_in_176	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 176

Port	Direction	Protocol	Description
ppi_in_177	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 177
ppi_in_178	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 178
ppi_in_179	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 179
ppi_in_17	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 17.
ppi_in_180	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 180
ppi_in_181	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 181
ppi_in_182	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 182
ppi_in_183	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 183
ppi_in_184	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 184
ppi_in_185	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 185
ppi_in_186	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 186
ppi_in_187	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 187
ppi_in_188	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 188
ppi_in_189	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 189
ppi_in_18	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 18.
ppi_in_190	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 190
ppi_in_191	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 191
ppi_in_192	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 192
ppi_in_193	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 193
ppi_in_194	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 194
ppi_in_195	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 195
ppi_in_196	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 196
ppi_in_197	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 197
ppi_in_198	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 198
ppi_in_199	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 199
ppi_in_19	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 19.
ppi_in_1	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 1.
ppi_in_200	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 200
ppi_in_201	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 201
ppi_in_202	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 202
ppi_in_203	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 203
ppi_in_204	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 204
ppi_in_205	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 205
ppi_in_206	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 206
ppi_in_207	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 207
ppi_in_208	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 208
ppi_in_209	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 209
ppi_in_20	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 20.
ppi_in_210	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 210
ppi_in_211	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 211

Port	Direction	Protocol	Description
ppi_in_212	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 212
ppi_in_213	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 213
ppi_in_214	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 214
ppi_in_215	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 215
ppi_in_216	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 216
ppi_in_217	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 217
ppi_in_218	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 218
ppi_in_219	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 219
ppi_in_21	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 21.
ppi_in_220	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 220
ppi_in_221	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 221
ppi_in_222	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 222
ppi_in_223	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 223
ppi_in_224	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 224
ppi_in_225	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 225
ppi_in_226	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 226
ppi_in_227	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 227
ppi_in_228	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 228
ppi_in_229	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 229
ppi_in_22	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 22.
ppi_in_230	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 230
ppi_in_231	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 231
ppi_in_232	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 232
ppi_in_233	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 233
ppi_in_234	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 234
ppi_in_235	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 235
ppi_in_236	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 236
ppi_in_237	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 237
ppi_in_238	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 238
ppi_in_239	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 239
ppi_in_23	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 23.
ppi_in_240	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 240
ppi_in_241	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 241
ppi_in_242	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 242
ppi_in_243	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 243
ppi_in_244	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 244
ppi_in_245	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 245
ppi_in_246	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 246
ppi_in_247	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 247
ppi_in_248	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 248

Port	Direction	Protocol	Description
ppi_in_249	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 249
ppi_in_24	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 24.
ppi_in_250	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 250
ppi_in_251	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 251
ppi_in_252	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 252
ppi_in_253	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 253
ppi_in_254	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 254
ppi_in_255	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 255
ppi_in_25	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 25.
ppi_in_26	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 26.
ppi_in_27	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 27.
ppi_in_28	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 28.
ppi_in_29	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 29.
ppi_in_2	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 2.
ppi_in_30	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 30.
ppi_in_31	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 31.
ppi_in_32	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 32.
ppi_in_33	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 33.
ppi_in_34	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 34.
ppi_in_35	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 35.
ppi_in_36	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 36.
ppi_in_37	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 37.
ppi_in_38	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 38.
ppi_in_39	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 39.
ppi_in_3	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 3.
ppi_in_40	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 40.
ppi_in_41	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 41.
ppi_in_42	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 42.
ppi_in_43	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 43.
ppi_in_44	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 44.
ppi_in_45	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 45.
ppi_in_46	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 46.
ppi_in_47	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 47.
ppi_in_48	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 48.
ppi_in_49	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 49.
ppi_in_4	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 4.
ppi_in_50	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 50.
ppi_in_51	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 51.
ppi_in_52	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 52.
ppi_in_53	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 53.



Port	Direction	Protocol	Description
ppi_in_54	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 54.
ppi_in_55	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 55.
ppi_in_56	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 56.
ppi_in_57	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 57.
ppi_in_58	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 58.
ppi_in_59	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 59.
ppi_in_5	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 5.
ppi_in_60	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 60.
ppi_in_61	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 61.
ppi_in_62	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 62.
ppi_in_63	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 63.
ppi_in_64	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 64.
ppi_in_65	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 65.
ppi_in_66	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 66.
ppi_in_67	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 67.
ppi_in_68	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 68.
ppi_in_69	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 69.
ppi_in_6	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 6.
ppi_in_70	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 70.
ppi_in_71	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 71.
ppi_in_72	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 72.
ppi_in_73	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 73.
ppi_in_74	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 74.
ppi_in_75	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 75.
ppi_in_76	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 76.
ppi_in_77	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 77.
ppi_in_78	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 78.
ppi_in_79	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 79.
ppi_in_7	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 7.
ppi_in_80	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 80.
ppi_in_81	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 81.
ppi_in_82	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 82.
ppi_in_83	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 83.
ppi_in_84	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 84.
ppi_in_85	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 85.
ppi_in_86	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 86.
ppi_in_87	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 87.
ppi_in_88	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 88.
ppi_in_89	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 89.
ppi_in_8	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 8.



Port	Direction	Protocol	Description
ppi_in_90	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 90
ppi_in_91	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 91
ppi_in_92	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 92
ppi_in_93	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 93
ppi_in_94	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 94
ppi_in_95	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 95
ppi_in_96	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 96
ppi_in_97	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 97
ppi_in_98	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 98
ppi_in_99	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 99
ppi_in_9	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 9.
pvbus_filtermiss_m	master	PVBus	Memory bus out. Transactions not filtered by the component.
pvbus_m	master	PVBus	Memory bus out: transactions generated by the IRI.
pvbus_s	slave	PVBus	Memory bus in: memory-mapped register accesses are accepted on this interface.
redistributor_m	master	GICv3Comms	Input from and output to CPU interface.
reset	slave	Signal	Reset.
spi_in	slave	Signal	Shared peripheral interrupts.
wake_request	master	Signal	Power management outputs.

## Parameters for GIC700\_Filter

### CPU-affinities

A comma separated list of dotted quads containing the affinities of all PEs connected to this IRI. If CPU-affinities-file is specified, this parameter is ignored.

Type: `string`

Default value: `N/A`

### CPU-affinities-file

A file containing comma separated list of dotted quads containing the affinities of all PEs connected to this IRI. If this parameter is specified, CPU-affinities parameter will be ignored even when it is given.

Type: `string`

Default value: `N/A`

### GICD\_CTLR-DS-1-means-secure-only

If GICD\_CTLR.DS=1, GICD supports a single security state which is secure if this is true, otherwise is non-secure.

Type: `bool`

Default value: `false`

### **GICD\_TYPER2**

GICD\_TYPER2 value containing VID and VIL to define the width of vPEID for GICv4.1.

Type: `uint32_t`

Default value: 0

### **GICR-clear-enable-supported**

When true, this sets the value of the RO bit GICR\_CTLR.CES with the value of the parameter `allow-LPIEN-clear`, making it visible to software.

Type: `bool`

Default value: `false`

### **GICR-invalidate-registers-implemented**

When true, the registers GICR\_INVLPIR, GICR\_INVALLR and GICR\_SYNCNR are implemented.

Type: `bool`

Default value: `false`

### **IIDR**

GICD\_IIDR, GICR\_IIDR and GITS\_IIDR value. Defaults to the latest modelled revision.

Type: `uint32_t`

Default value: `0x0404043B`

### **ITS-ID-bits**

Number of interrupt bits supported by ITS.

Type: `uint8_t`

Default value: 16

### **ITS-collection-ID-bits**

Number of collection bits supported by ITS (optional parameter, 0 => 16bits support and GITS\_TYPER.CIL=0).

Type: `uint8_t`

Default value: 8

**ITS-count**

Number of Interrupt Translation Services to be instantiated (0=none).

Type: `uint8_t`

Default value: 1

**ITS-device-bits**

Number of bits supported for ITS device IDs.

Type: `uint8_t`

Default value: 16

**ITS-enable-itt-address-verification**

If true, a transaction will be sent to ITT Address for verification.

Type: `bool`

Default value: `false`

**ITS-hardware-collection-count**

Number of hardware collections held exclusively in the ITS.

Type: `uint8_t`

Default value: 0

**ITS-shared-vPE-table**

Number of affinity levels to which the vPE configuration table is shared. This parameter is valid when `has-gicv4.1` is true.

Type: `uint8_t`

Default value: 0

**ITS-vmovp-bit**

Device supports software issuing a VMOVP to only one of the ITSs that has a mapping for a vPE. The device itself ensures synchronization of the VMOVP command across all ITSs that have mapping for that vPE.

Type: `bool`

Default value: `false`

**PPI-count**

Selects the number of PPI available for each PE

**8**

id22-27,29,30

**12**

id 20-31

**16**

id 16-31.

Type: `int`

Default value: 16

**SPI-blocks**

Number of SPI blocks supported by the IRI, each block contains 32 SPIs.

Type: `int`

Default value: 62

**add-output-cpu-wake-request-signal-from-redistributor**If true, the redistributor will have the output signal `cpu_wake_request` from GIC to DSU and if false, the signals are not added to the redistributor.Type: `bool`Default value: `false`**affinity-width**

A dotted quad indicating the bitwidth of fields at each affinity level.

Type: `string`

Default value: "4.8.8.8"

**allow-LPIEN-clear**

Allow RW behaviour on GICR\_CTLR.LPIEN instead of set once.

Type: `bool`Default value: `true`**chip-count**

The total number of chips supported.

Type: `int`

Default value: 16

**chip-id**

Chip ID for multichip operation.

Type: `uint8_t`

Default value: 0

**chip-select-affinity-level**

Affinity level 2 or 3 can be used for chip select.

Type: `int`

Default value: 3

**clear-ISPENDR-bit-for-level-sensitive-interrupt-when-acknowledged**

If true, acknowledgement of a level sensitive interrupt clears the corresponding bit in the ISPENDR register.

Type: `bool`

Default value: `false`

**common-lpi-configuration**

Describes which re-distributors share (and must be configured with the same) LPI configuration table as described in GICR\_TYPER( 0:All, 1:A.x.x.x, 2:A.B.x.x, 3:A.B.C.x).

Type: `int`

Default value: 0

**common-vPE-table-affinity**

Affinity value list in the form of 'a.b.c.d, e.f.g.h, etc' given to the ITS(s), where a.b.c.d corresponds to ITS0, e.f.g.h corresponds to ITS1 and so on. Under an affinity value the vPE configuration table is shared among redistributors where the level to be shared is defined by ITS-shared-vPE-table. This parameter is valid when `has-gicv4.1` is true.

Type: `string`

Default value: N/A

**consolidators**

Specify consolidators' base addresses, interrupt line counts and base interrupt IDs, in the form:

```
'baseAddr0:itlineCount0:baseINTID0, baseAddr1:itlineCount1:baseINTID1'
```

For example:

```
'0x3f100000:64:4096, 0x3f200000:64:4224'
```

The consolidators' count is inferred from the list (maximum of 4). If not specified, the component contains no consolidators.

Type: `string`

Default value: `""`

### **cross-chip-AMBA-is-ACE**

Indicates the AMBA protocol that the cross-chip interface uses. If true, the cross-chip interface uses the ACE5-Lite protocol. Otherwise, it uses the AXI5-Stream protocol.

Type: `bool`

Default value: `false`

### **enable-local-cross-chip-addressing**

If true, each distributor in a multichip system will use routing table address (ADDR) values programmed by local software, and will not be overwritten by the default chip. Otherwise, all the distributors will share the same routing table address values programmed in the default chip.

Type: `bool`

Default value: `false`

### **enable-multiple-views-feature**

If true, multiple view feature will provide multiple programming views which can be used by multiple hypervisors.

Type: `bool`

Default value: `false`

### **enabled**

Enable GICv3 functionality; when false the component is inactive.

Type: `bool`

Default value: `true`

### **extended-ppi-count**

Number of extended PPI supported.

Type: `unsigned`

Default value: 64

**gicp-allow-ns-reset**

If true, non-secure read/write access to GICP register is allowed at reset. Not allowed otherwise. This emulates gicp\_allow\_ns tie-off signal.

Type: bool

Default value: true

**gict-allow-ns-reset**

If true, non-secure read/write access to GICT register is allowed at reset. Not allowed otherwise. This emulates gict\_allow\_ns tie-off signal.

Type: bool

Default value: true

**has-gicv4.1**

Enable GICv4.1 functionality; when false the component is inactive.

Type: bool

Default value: true

**has-two-security-states**

If true, has two security states.

Type: bool

Default value: true

**has\_VPENDBASER-dirty-flag-on-load**

GICR\_VPENDBASER.Dirty reflects transient loading state when valid=1.

Type: bool

Default value: false

**has\_nmi**

Enable support for Non-maskable Interrupts (NMIs).

Type: bool

Default value: false

**max-cores-supported-by-GCI**

GCI can support 1-64 cores. Maximum supported cores in GCI is configurable by device.

Type: `int`

Default value: `64`

**max-pe-on-chip**

Maximum number of cores on any single chip. This will be used to identify the target chip and core when multichip operation is enabled.

Type: `int`

Default value: `4`

**multichip-threaded-dgi**

Enable sending of multichip DGI messages in a separate thread, when multichip operation is enabled.

Type: `bool`

Default value: `true`

**output\_attributes**

User-defined transform to be applied to bus attributes like ManagerID, ExtendedID or UserFlags. Currently, only works for MPAM Attributes encoding into bus attributes.

Type: `string`

Default value: `"ExtendedID[62:55]=MPAM_PMG, ExtendedID[54:39]=MPAM_PARTID, ExtendedID[38]=MPAM_NS"`

**print-memory-map**

Print memory map to stdout.

Type: `bool`

Default value: `false`

**prog-mpidr**

Whether software or hardware can remove cores from a GIC configuration.

**0**

none

**1**

prog - Secure software to remove cores during the boot up of a system.



## 2

strap - enables hardware to remove cores as GIC exits reset.

Type: `unsigned`

Default value: 0

### **redistributor-group**

Redistributor grouping information with affinity as JSON :

```
{
  "0": [
    "0.0.0.0",
    "0.0.0.1"
  ],
  "1": [
    "0.0.1.0",
    "0.0.1.1"
  ]
}
```

where RD with quad 0.0.0.0 and 0.0.0.1 belongs to RD group 0. All the RDs belong to one group when this parameter is not given.

Type: `string`

Default value: N/A

### **redistributor-group-file**

File path to redistributor grouping information with affinity as JSON.

The file uses the same format as `redistributor-group` parameter.

Type: `string`

Default value: N/A

### **redistributor-power-managed-by-pwrr**

GIC-700 redistributor power management is done by updating GICR\_PWRR register.

Type: `bool`

Default value: `true`

### **reg-base**

GIC-700 base address for memory-mapped register.

Type: `uint64_t`

Default value: `0x2c010000`

**reg-base-per-redistributor**

Base address for each redistributor in the form:

```
0.0.0.0=0x2c010000, 0.0.0.1=0x2c020000
```

All redistributors must be specified and this overrides the `reg-base` parameter (except that `reg-base` will still be used for the top-level redistributor). If `reg-base-per-redistributor-file` is specified, this parameter is ignored.

Type: `string`

Default value: `""`

**view-id-bits-offset**

The offset of the view-id bits in the address. Effective only when multiple views feature is enabled. Accepted values will depend on the configuration affecting the GIC memory footprint, such as the input to the parameter 'CPU-affinities'. For example, for a 32MB-sized view, the value of this parameter should be 25.

Type: `int`

Default value: `0`

## 3.196 GIC720AE

Defined in `LISA/GIC720AE.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r2p1	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

**Changes in 11.31.15**

Model quality level changes:

From	To
Preliminary support	Full support

IP revision changes:

From	To
N/A	r2p1

## About GIC720AE

The model has the same functionality as [GIC700](#), but in addition supports the following AE-specific features:

- GIC FMU

It has the same limitations as GIC700.

## Iris and MTI instances for GIC720AE

This model has the following Iris instances:

Name	Instance type
GIC720AE	GIC720AE
GIC720AE.GICV3_ProtocolChecker	GICv3ProtocolChecker
GIC720AE.ITS0	GICv4InterruptTranslationService
GIC720AE.ITS0.ExportTest.GIC720AE.ITS0.pvbus_m[0].pvbusmaster	PVBusMaster
GIC720AE.ITS0.bus_subordinate	PVBusSlave
GIC720AE.fmu	FMU
GIC720AE.fmu.pvbus_slave	PVBusSlave
GIC720AE.rd_0	GICv4RedistributorInternal
GIC720AE.rd_0_0	GICv4RedistributorInternal
GIC720AE.rd_0_0_0	GICv4RedistributorInternal
GIC720AE.rd_0_0_0_0	GICv3Redistributor
GIC720AE.rd_0_0_0_0.ExportTest.GIC720AE.rd_0_0_0_0.pvbus_m[0].pvbusmaster	PVBusMaster
GIC720AE.rd_t1	GICv3Distributor
GIC720AE.rd_t1.ExportTest.GIC720AE.rd_t1.pvbus_m[0].pvbusmaster	PVBusMaster

This model has the following MTI trace components:

Name	Component type
GIC720AE.GICV3_ProtocolChecker	GICv3ProtocolChecker
GIC720AE.ITS0	GICv4InterruptTranslationService
GIC720AE.ITS0.ExportTest.GIC720AE.ITS0.pvbus_m[0].pvbusmaster	PVBusMaster
GIC720AE.ITS0.bus_subordinate	PVBusSlave
GIC720AE.fmu	FMU
GIC720AE.fmu.pvbus_slave	PVBusSlave
GIC720AE.rd_0	GICv4RedistributorInternal
GIC720AE.rd_0_0	GICv4RedistributorInternal
GIC720AE.rd_0_0_0	GICv4RedistributorInternal
GIC720AE.rd_0_0_0_0	GICv3Redistributor
GIC720AE.rd_0_0_0_0.ExportTest.GIC720AE.rd_0_0_0_0.pvbus_m[0].pvbusmaster	PVBusMaster
GIC720AE.rd_t1	GICv3Distributor
GIC720AE.rd_t1.ExportTest.GIC720AE.rd_t1.pvbus_m[0].pvbusmaster	PVBusMaster

## Ports for GIC720AE

Port	Direction	Protocol	Description
apb_bus	slave	PVBus	FMU signals
axi_stream_msi_s	slave	PVBus	AXI Stream Interface ports for MSI (message-signalled interrupt) translation, in GIC-720AE. Typically the SMMU's TCU connects to this port for MSI.
chip_id	slave	Value	chip_id port for multichip operation
cpu_active_s	slave	Signal	CPUActive pins.
cpu_wake_request	master	Signal	-
err_int	master	Signal	Error Recovery Interrupt
extended_ppi_in_0	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 0.
extended_ppi_in_100	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 100
extended_ppi_in_101	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 101
extended_ppi_in_102	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 102
extended_ppi_in_103	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 103
extended_ppi_in_104	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 104
extended_ppi_in_105	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 105
extended_ppi_in_106	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 106
extended_ppi_in_107	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 107
extended_ppi_in_108	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 108
extended_ppi_in_109	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 109
extended_ppi_in_110	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 110
extended_ppi_in_111	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 111
extended_ppi_in_112	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 112
extended_ppi_in_113	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 113
extended_ppi_in_114	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 114
extended_ppi_in_115	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 115
extended_ppi_in_116	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 116
extended_ppi_in_117	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 117
extended_ppi_in_118	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 118
extended_ppi_in_119	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 119
extended_ppi_in_120	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 120
extended_ppi_in_121	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 121
extended_ppi_in_122	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 122
extended_ppi_in_123	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 123
extended_ppi_in_124	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 124
extended_ppi_in_125	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 125
extended_ppi_in_126	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 126
extended_ppi_in_127	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 127
extended_ppi_in_128	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 128

Port	Direction	Protocol	Description
extended_ppi_in_129	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 129
extended_ppi_in_12	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 12.
extended_ppi_in_130	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 130
extended_ppi_in_131	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 131
extended_ppi_in_132	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 132
extended_ppi_in_133	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 133
extended_ppi_in_134	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 134
extended_ppi_in_135	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 135
extended_ppi_in_136	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 136
extended_ppi_in_137	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 137
extended_ppi_in_138	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 138
extended_ppi_in_139	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 139
extended_ppi_in_13	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 13.
extended_ppi_in_140	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 140
extended_ppi_in_141	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 141
extended_ppi_in_142	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 142
extended_ppi_in_143	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 143
extended_ppi_in_144	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 144
extended_ppi_in_145	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 145
extended_ppi_in_146	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 146
extended_ppi_in_147	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 147
extended_ppi_in_148	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 148
extended_ppi_in_149	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 149
extended_ppi_in_14	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 14.
extended_ppi_in_150	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 150
extended_ppi_in_151	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 151
extended_ppi_in_152	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 152
extended_ppi_in_153	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 153
extended_ppi_in_154	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 154
extended_ppi_in_155	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 155
extended_ppi_in_156	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 156
extended_ppi_in_157	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 157
extended_ppi_in_158	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 158
extended_ppi_in_159	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 159
extended_ppi_in_15	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 15.
extended_ppi_in_160	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 160
extended_ppi_in_161	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 161
extended_ppi_in_162	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 162
extended_ppi_in_163	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 163
extended_ppi_in_164	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 164

Port	Direction	Protocol	Description
extended_ppi_in_165	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 165
extended_ppi_in_166	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 166
extended_ppi_in_167	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 167
extended_ppi_in_168	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 168
extended_ppi_in_169	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 169
extended_ppi_in_16	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 16.
extended_ppi_in_170	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 170
extended_ppi_in_171	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 171
extended_ppi_in_172	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 172
extended_ppi_in_173	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 173
extended_ppi_in_174	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 174
extended_ppi_in_175	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 175
extended_ppi_in_176	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 176
extended_ppi_in_177	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 177
extended_ppi_in_178	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 178
extended_ppi_in_179	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 179
extended_ppi_in_17	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 17.
extended_ppi_in_180	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 180
extended_ppi_in_181	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 181
extended_ppi_in_182	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 182
extended_ppi_in_183	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 183
extended_ppi_in_184	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 184
extended_ppi_in_185	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 185
extended_ppi_in_186	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 186
extended_ppi_in_187	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 187
extended_ppi_in_188	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 188
extended_ppi_in_189	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 189
extended_ppi_in_18	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 18.
extended_ppi_in_190	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 190
extended_ppi_in_191	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 191
extended_ppi_in_192	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 192
extended_ppi_in_193	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 193
extended_ppi_in_194	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 194
extended_ppi_in_195	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 195
extended_ppi_in_196	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 196
extended_ppi_in_197	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 197
extended_ppi_in_198	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 198
extended_ppi_in_199	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 199
extended_ppi_in_19	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 19.
extended_ppi_in_1	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 1.

Port	Direction	Protocol	Description
extended_ppi_in_200	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 200
extended_ppi_in_201	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 201
extended_ppi_in_202	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 202
extended_ppi_in_203	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 203
extended_ppi_in_204	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 204
extended_ppi_in_205	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 205
extended_ppi_in_206	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 206
extended_ppi_in_207	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 207
extended_ppi_in_208	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 208
extended_ppi_in_209	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 209
extended_ppi_in_20	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 20.
extended_ppi_in_210	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 210
extended_ppi_in_211	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 211
extended_ppi_in_212	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 212
extended_ppi_in_213	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 213
extended_ppi_in_214	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 214
extended_ppi_in_215	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 215
extended_ppi_in_216	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 216
extended_ppi_in_217	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 217
extended_ppi_in_218	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 218
extended_ppi_in_219	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 219
extended_ppi_in_21	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 21.
extended_ppi_in_220	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 220
extended_ppi_in_221	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 221
extended_ppi_in_222	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 222
extended_ppi_in_223	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 223
extended_ppi_in_224	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 224
extended_ppi_in_225	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 225
extended_ppi_in_226	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 226
extended_ppi_in_227	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 227
extended_ppi_in_228	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 228
extended_ppi_in_229	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 229
extended_ppi_in_22	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 22.
extended_ppi_in_230	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 230
extended_ppi_in_231	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 231
extended_ppi_in_232	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 232
extended_ppi_in_233	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 233
extended_ppi_in_234	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 234
extended_ppi_in_235	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 235
extended_ppi_in_236	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 236



Port	Direction	Protocol	Description
extended_ppi_in_237	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 237
extended_ppi_in_238	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 238
extended_ppi_in_239	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 239
extended_ppi_in_23	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 23.
extended_ppi_in_240	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 240
extended_ppi_in_241	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 241
extended_ppi_in_242	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 242
extended_ppi_in_243	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 243
extended_ppi_in_244	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 244
extended_ppi_in_245	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 245
extended_ppi_in_246	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 246
extended_ppi_in_247	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 247
extended_ppi_in_248	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 248
extended_ppi_in_249	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 249
extended_ppi_in_24	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 24.
extended_ppi_in_250	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 250
extended_ppi_in_251	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 251
extended_ppi_in_252	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 252
extended_ppi_in_253	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 253
extended_ppi_in_254	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 254
extended_ppi_in_255	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 255
extended_ppi_in_25	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 25.
extended_ppi_in_26	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 26.
extended_ppi_in_27	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 27.
extended_ppi_in_28	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 28.
extended_ppi_in_29	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 29.
extended_ppi_in_2	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 2.
extended_ppi_in_30	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 30.
extended_ppi_in_31	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 31.
extended_ppi_in_32	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 32.
extended_ppi_in_33	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 33.
extended_ppi_in_34	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 34.
extended_ppi_in_35	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 35.
extended_ppi_in_36	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 36.
extended_ppi_in_37	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 37.
extended_ppi_in_38	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 38.
extended_ppi_in_39	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 39.
extended_ppi_in_3	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 3.
extended_ppi_in_40	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 40.
extended_ppi_in_41	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 41.



Port	Direction	Protocol	Description
extended_ppi_in_42	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 42.
extended_ppi_in_43	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 43.
extended_ppi_in_44	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 44.
extended_ppi_in_45	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 45.
extended_ppi_in_46	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 46.
extended_ppi_in_47	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 47.
extended_ppi_in_48	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 48.
extended_ppi_in_49	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 49.
extended_ppi_in_4	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 4.
extended_ppi_in_50	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 50.
extended_ppi_in_51	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 51.
extended_ppi_in_52	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 52.
extended_ppi_in_53	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 53.
extended_ppi_in_54	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 54.
extended_ppi_in_55	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 55.
extended_ppi_in_56	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 56.
extended_ppi_in_57	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 57.
extended_ppi_in_58	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 58.
extended_ppi_in_59	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 59.
extended_ppi_in_5	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 5.
extended_ppi_in_60	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 60.
extended_ppi_in_61	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 61.
extended_ppi_in_62	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 62.
extended_ppi_in_63	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 63.
extended_ppi_in_64	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 64.
extended_ppi_in_65	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 65.
extended_ppi_in_66	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 66.
extended_ppi_in_67	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 67.
extended_ppi_in_68	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 68.
extended_ppi_in_69	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 69.
extended_ppi_in_6	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 6.
extended_ppi_in_70	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 70.
extended_ppi_in_71	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 71.
extended_ppi_in_72	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 72.
extended_ppi_in_73	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 73.
extended_ppi_in_74	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 74.
extended_ppi_in_75	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 75.
extended_ppi_in_76	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 76.
extended_ppi_in_77	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 77.
extended_ppi_in_78	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 78.

Port	Direction	Protocol	Description
extended_ppi_in_79	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 79
extended_ppi_in_7	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 7.
extended_ppi_in_80	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 80
extended_ppi_in_81	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 81
extended_ppi_in_82	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 82
extended_ppi_in_83	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 83
extended_ppi_in_84	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 84
extended_ppi_in_85	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 85
extended_ppi_in_86	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 86
extended_ppi_in_87	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 87
extended_ppi_in_88	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 88
extended_ppi_in_89	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 89
extended_ppi_in_8	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 8.
extended_ppi_in_90	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 90
extended_ppi_in_91	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 91
extended_ppi_in_92	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 92
extended_ppi_in_93	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 93
extended_ppi_in_94	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 94
extended_ppi_in_95	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 95
extended_ppi_in_96	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 96
extended_ppi_in_97	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 97
extended_ppi_in_98	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 98
extended_ppi_in_99	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 99
extended_ppi_in_9	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 9.
extended_spi_in	slave	Signal	Extended Shared peripheral interrupts.
fault_int	master	Signal	RAS Interrupt signals Fault Handling Interrupt
fmu_cri	master	Signal	Critical Interrupt
fmu_eri	master	Signal	Error recovery Interrupt
icdrdt_out	master	PVBus	AXI Stream Interface. This is used only when local cross chip addressing is enabled at the moment
icrdt_in	slave	PVBus	-
po_reset	slave	Signal	Resets.
ppi_in_0	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 0.
ppi_in_100	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 100
ppi_in_101	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 101
ppi_in_102	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 102
ppi_in_103	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 103
ppi_in_104	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 104
ppi_in_105	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 105
ppi_in_106	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 106
ppi_in_107	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 107

Port	Direction	Protocol	Description
ppi_in_108	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 108
ppi_in_109	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 109
ppi_in_10	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 10.
ppi_in_110	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 110
ppi_in_111	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 111
ppi_in_112	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 112
ppi_in_113	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 113
ppi_in_114	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 114
ppi_in_115	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 115
ppi_in_116	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 116
ppi_in_117	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 117
ppi_in_118	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 118
ppi_in_119	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 119
ppi_in_11	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 11.
ppi_in_120	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 120
ppi_in_121	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 121
ppi_in_122	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 122
ppi_in_123	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 123
ppi_in_124	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 124
ppi_in_125	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 125
ppi_in_126	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 126
ppi_in_127	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 127
ppi_in_128	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 128
ppi_in_129	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 129
ppi_in_12	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 12.
ppi_in_130	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 130
ppi_in_131	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 131
ppi_in_132	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 132
ppi_in_133	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 133
ppi_in_134	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 134
ppi_in_135	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 135
ppi_in_136	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 136
ppi_in_137	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 137
ppi_in_138	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 138
ppi_in_139	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 139
ppi_in_13	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 13.
ppi_in_140	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 140
ppi_in_141	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 141
ppi_in_142	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 142
ppi_in_143	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 143

Port	Direction	Protocol	Description
ppi_in_144	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 144
ppi_in_145	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 145
ppi_in_146	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 146
ppi_in_147	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 147
ppi_in_148	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 148
ppi_in_149	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 149
ppi_in_14	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 14.
ppi_in_150	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 150
ppi_in_151	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 151
ppi_in_152	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 152
ppi_in_153	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 153
ppi_in_154	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 154
ppi_in_155	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 155
ppi_in_156	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 156
ppi_in_157	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 157
ppi_in_158	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 158
ppi_in_159	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 159
ppi_in_15	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 15.
ppi_in_160	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 160
ppi_in_161	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 161
ppi_in_162	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 162
ppi_in_163	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 163
ppi_in_164	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 164
ppi_in_165	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 165
ppi_in_166	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 166
ppi_in_167	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 167
ppi_in_168	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 168
ppi_in_169	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 169
ppi_in_16	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 16.
ppi_in_170	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 170
ppi_in_171	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 171
ppi_in_172	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 172
ppi_in_173	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 173
ppi_in_174	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 174
ppi_in_175	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 175
ppi_in_176	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 176
ppi_in_177	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 177
ppi_in_178	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 178
ppi_in_179	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 179
ppi_in_17	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 17.

Port	Direction	Protocol	Description
ppi_in_180	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 180
ppi_in_181	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 181
ppi_in_182	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 182
ppi_in_183	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 183
ppi_in_184	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 184
ppi_in_185	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 185
ppi_in_186	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 186
ppi_in_187	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 187
ppi_in_188	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 188
ppi_in_189	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 189
ppi_in_18	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 18.
ppi_in_190	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 190
ppi_in_191	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 191
ppi_in_192	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 192
ppi_in_193	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 193
ppi_in_194	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 194
ppi_in_195	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 195
ppi_in_196	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 196
ppi_in_197	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 197
ppi_in_198	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 198
ppi_in_199	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 199
ppi_in_19	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 19.
ppi_in_1	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 1.
ppi_in_200	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 200
ppi_in_201	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 201
ppi_in_202	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 202
ppi_in_203	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 203
ppi_in_204	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 204
ppi_in_205	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 205
ppi_in_206	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 206
ppi_in_207	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 207
ppi_in_208	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 208
ppi_in_209	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 209
ppi_in_20	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 20.
ppi_in_210	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 210
ppi_in_211	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 211
ppi_in_212	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 212
ppi_in_213	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 213
ppi_in_214	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 214
ppi_in_215	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 215

Port	Direction	Protocol	Description
ppi_in_216	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 216
ppi_in_217	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 217
ppi_in_218	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 218
ppi_in_219	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 219
ppi_in_21	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 21.
ppi_in_220	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 220
ppi_in_221	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 221
ppi_in_222	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 222
ppi_in_223	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 223
ppi_in_224	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 224
ppi_in_225	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 225
ppi_in_226	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 226
ppi_in_227	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 227
ppi_in_228	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 228
ppi_in_229	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 229
ppi_in_22	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 22.
ppi_in_230	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 230
ppi_in_231	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 231
ppi_in_232	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 232
ppi_in_233	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 233
ppi_in_234	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 234
ppi_in_235	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 235
ppi_in_236	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 236
ppi_in_237	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 237
ppi_in_238	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 238
ppi_in_239	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 239
ppi_in_23	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 23.
ppi_in_240	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 240
ppi_in_241	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 241
ppi_in_242	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 242
ppi_in_243	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 243
ppi_in_244	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 244
ppi_in_245	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 245
ppi_in_246	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 246
ppi_in_247	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 247
ppi_in_248	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 248
ppi_in_249	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 249
ppi_in_24	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 24.
ppi_in_250	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 250
ppi_in_251	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 251

Port	Direction	Protocol	Description
ppi_in_252	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 252
ppi_in_253	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 253
ppi_in_254	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 254
ppi_in_255	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 255
ppi_in_25	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 25.
ppi_in_26	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 26.
ppi_in_27	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 27.
ppi_in_28	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 28.
ppi_in_29	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 29.
ppi_in_2	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 2.
ppi_in_30	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 30.
ppi_in_31	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 31.
ppi_in_32	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 32.
ppi_in_33	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 33.
ppi_in_34	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 34.
ppi_in_35	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 35.
ppi_in_36	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 36.
ppi_in_37	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 37.
ppi_in_38	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 38.
ppi_in_39	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 39.
ppi_in_3	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 3.
ppi_in_40	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 40.
ppi_in_41	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 41.
ppi_in_42	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 42.
ppi_in_43	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 43.
ppi_in_44	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 44.
ppi_in_45	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 45.
ppi_in_46	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 46.
ppi_in_47	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 47.
ppi_in_48	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 48.
ppi_in_49	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 49.
ppi_in_4	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 4.
ppi_in_50	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 50.
ppi_in_51	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 51.
ppi_in_52	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 52.
ppi_in_53	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 53.
ppi_in_54	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 54.
ppi_in_55	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 55.
ppi_in_56	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 56.
ppi_in_57	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 57.



Port	Direction	Protocol	Description
ppi_in_58	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 58.
ppi_in_59	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 59.
ppi_in_5	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 5.
ppi_in_60	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 60.
ppi_in_61	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 61.
ppi_in_62	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 62.
ppi_in_63	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 63.
ppi_in_64	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 64.
ppi_in_65	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 65.
ppi_in_66	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 66.
ppi_in_67	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 67.
ppi_in_68	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 68.
ppi_in_69	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 69.
ppi_in_6	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 6.
ppi_in_70	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 70.
ppi_in_71	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 71.
ppi_in_72	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 72.
ppi_in_73	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 73.
ppi_in_74	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 74.
ppi_in_75	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 75.
ppi_in_76	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 76.
ppi_in_77	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 77.
ppi_in_78	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 78.
ppi_in_79	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 79.
ppi_in_7	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 7.
ppi_in_80	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 80.
ppi_in_81	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 81.
ppi_in_82	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 82.
ppi_in_83	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 83.
ppi_in_84	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 84.
ppi_in_85	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 85.
ppi_in_86	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 86.
ppi_in_87	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 87.
ppi_in_88	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 88.
ppi_in_89	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 89.
ppi_in_8	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 8.
ppi_in_90	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 90.
ppi_in_91	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 91.
ppi_in_92	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 92.
ppi_in_93	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 93.



Port	Direction	Protocol	Description
ppi_in_94	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 94
ppi_in_95	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 95
ppi_in_96	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 96
ppi_in_97	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 97
ppi_in_98	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 98
ppi_in_99	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 99
ppi_in_9	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 9.
pvbus_m	master	PVBus	Memory bus out: transactions generated by the IRI.
pvbus_s	slave	PVBus	Memory bus in: memory-mapped register accesses are accepted on this interface.
redistributor_m	master	GICv3Comms	Input from and output to CPU interface.
reset	slave	Signal	Resets.
spi_in	slave	Signal	Shared peripheral interrupts.
wake_request	master	Signal	Power management outputs.

## Parameters for GIC720AE

### CPU-affinities

A comma separated list of dotted quads containing the affinities of all PEs connected to this IRI. If CPU-affinities-file is specified, this parameter is ignored.

Type: string

Default value: N/A

### CPU-affinities-file

A file containing comma separated list of dotted quads containing the affinities of all PEs connected to this IRI. If this parameter is specified, CPU-affinities parameter will be ignored even when it is given.

Type: string

Default value: N/A

### GICD\_CTLR-DS-1-means-secure-only

If GICD\_CTLR.DS=1, GICD supports a single security state which is secure if this is true, otherwise is non-secure.

Type: bool

Default value: false

### GICD\_TYPER2

GICD\_TYPER2 value containing VID and VIL to define the width of vPEID for GICv4.1.

Type: `uint32_t`

Default value: 0

### **GICR-clear-enable-supported**

When true, this sets the value of the RO bit `GICR_CTLR.CES` with the value of the parameter `allow-LPIEN-clear`, making it visible to software.

Type: `bool`

Default value: `false`

### **GICR-invalidate-registers-implemented**

When true, the registers `GICR_INVLPIR`, `GICR_INVALLR` and `GICR_SYNCR` are implemented.

Type: `bool`

Default value: `false`

### **IIDR**

`GICD_IIDR`, `GICR_IIDR` and `GITS_IIDR` value. Defaults to the latest modelled revision.

Type: `uint32_t`

Default value: `0x0702143B`

### **ITS-ID-bits**

Number of interrupt bits supported by ITS.

Type: `uint8_t`

Default value: 16

### **ITS-collection-ID-bits**

Number of collection bits supported by ITS (optional parameter, 0 => 16bits support and `GITS_TYPER.CIL=0`).

Type: `uint8_t`

Default value: 8

### **ITS-count**

Number of Interrupt Translation Services to be instantiated (0=none).

Type: `uint8_t`

Default value: 1

**ITS-device-bits**

Number of bits supported for ITS device IDs.

Type: `uint8_t`

Default value: 16

**ITS-enable-itt-address-verification**

If true, a transaction will be sent to ITT Address for verification.

Type: `bool`

Default value: `false`

**ITS-hardware-collection-count**

Number of hardware collections held exclusively in the ITS.

Type: `uint8_t`

Default value: 0

**ITS-shared-vPE-table**

Number of affinity levels to which the vPE configuration table is shared. This parameter is valid when `has-gicv4.1` is true.

Type: `uint8_t`

Default value: 0

**ITS-vmovp-bit**

Device supports software issuing a VMOVP to only one of the ITSs that has a mapping for a vPE. The device itself ensures synchronization of the VMOVP command across all ITSs that have mapping for that vPE.

Type: `bool`

Default value: `false`

**PPI-count**

Selects the number of PPI available for each PE

**8**

id22-27,29,30

**12**

id 20-31

**16**

id 16-31.

Type: `int`

Default value: 16

### **SPI-blocks**

Number of SPI blocks supported by the IRI, each block contains 32 SPIs.

Type: `int`

Default value: 62

### **add-output-cpu-wake-request-signal-from-redistributor**

If true, the redistributor will have the output signal `cpu_wake_request` from GIC to DSU and if false, the signals are not added to the redistributor.

Type: `bool`

Default value: `false`

### **affinity-width**

A dotted quad indicating the bitwidth of fields at each affinity level.

Type: `string`

Default value: "4.8.8.8"

### **allow-LPIEN-clear**

Allow RW behaviour on GICR\_CTLR.LPIEN instead of set once.

Type: `bool`

Default value: `true`

### **chip-count**

The total number of chips supported.

Type: `int`

Default value: 16

### **chip-id**

Chip ID for multichip operation.

Type: `uint8_t`

Default value: 0

**chip-select-affinity-level**

Affinity level 2 or 3 can be used for chip select.

Type: `int`

Default value: `3`

**clear-ISPENDR-bit-for-level-sensitive-interrupt-when-acknowledged**

If true, acknowledgement of a level sensitive interrupt clears the corresponding bit in the ISPENDR register.

Type: `bool`

Default value: `false`

**common-lpi-configuration**

Describes which re-distributors share (and must be configured with the same) LPI configuration table as described in GICR\_TYPER( 0:All, 1:A.x.x.x, 2:A.B.x.x, 3:A.B.C.x).

Type: `int`

Default value: `0`

**common-vPE-table-affinity**

Affinity value list in the form of 'a.b.c.d, e.f.g.h, etc' given to the ITS(s), where a.b.c.d corresponds to ITS0, e.f.g.h corresponds to ITS1 and so on. Under an affinity value the vPE configuration table is shared among redistributors where the level to be shared is defined by ITS-shared-vPE-table. This parameter is valid when has-gicv4.1 is true.

Type: `string`

Default value: `N/A`

**consolidators**

Specify consolidators' base addresses, interrupt line counts and base interrupt IDs, in the form:

```
'baseAddr0:itlineCount0:baseINTID0, baseAddr1:itlineCount1:baseINTID1'
```

For example:

```
'0x3f100000:64:4096, 0x3f200000:64:4224'
```

The consolidators' count is inferred from the list (maximum of 4). If not specified, the component contains no consolidators.

Type: `string`

Default value: ""

**cross-chip-AMBA-is-ACE**

Indicates the AMBA protocol that the cross-chip interface uses. If true, the cross-chip interface uses the ACE5-Lite protocol. Otherwise, it uses the AXI5-Stream protocol. This parameter is meaningful only if enable-multichip-operation is set.

Type: `bool`

Default value: `false`

**enable-local-cross-chip-addressing**

If true, each distributor in a multichip system will use routing table address (ADDR) values programmed by local software, and will not be overwritten by the default chip. Otherwise, all the distributors will share the same routing table address values programmed in the default chip. This parameter is valid only if enable-multichip-operation is set.

Type: `bool`

Default value: `false`

**enable-multiple-views-feature**

If true, multiple view feature will provide multiple programming views which can be used by multiple hypervisors.

Type: `bool`

Default value: `false`

**enabled**

Enable GICv3 functionality; when false the component is inactive.

Type: `bool`

Default value: `true`

**extended-ppi-count**

Number of extended PPI supported.

Type: `unsigned`

Default value: 64

**fmv-blktype-num**

Number of stakeholder block types for FMU.

Type: `int`

Default value: 1

**gicp-allow-ns-reset**

If true, non-secure read/write access to GICP register is allowed at reset. Not allowed otherwise. This emulates gicp\_allow\_ns tie-off signal.

Type: bool

Default value: true

**gict-allow-ns-reset**

If true, non-secure read/write access to GICT register is allowed at reset. Not allowed otherwise. This emulates gict\_allow\_ns tie-off signal.

Type: bool

Default value: true

**has-gicv4.1**

Enable GICv4.1 functionality; when false the component is inactive.

Type: bool

Default value: true

**has-two-security-states**

If true, has two security states.

Type: bool

Default value: true

**has\_VPENDBASER-dirty-flag-on-load**

GICR\_VPENDBASER.Dirty reflects transient loading state when valid=1.

Type: bool

Default value: false

**has\_nmi**

Enable support for Non-maskable Interrupts (NMIs).

Type: bool

Default value: false

**max-cores-supported-by-GCI**

GCI can support 1-64 cores. Maximum supported cores in GCI is configurable by device.

Type: `int`

Default value: `64`

**max-pe-on-chip**

Maximum number of cores on any single chip. This will be used to identify the target chip and core when multichip operation is enabled.

Type: `int`

Default value: `4`

**multichip-threaded-dgi**

Enable sending of multichip DGI messages in a separate thread, when multichip operation is enabled.

Type: `bool`

Default value: `true`

**output\_attributes**

User-defined transform to be applied to bus attributes like ManagerID, ExtendedID or UserFlags. Currently, only works for MPAM Attributes encoding into bus attributes.

Type: `string`

Default value: `"ExtendedID[62:55]=MPAM_PMG, ExtendedID[54:39]=MPAM_PARTID, ExtendedID[38]=MPAM_NS"`

**print-memory-map**

Print memory map to stdout.

Type: `bool`

Default value: `false`

**prog-mpidr**

Whether software or hardware can remove cores from a GIC configuration.

**0**

none

**1**

prog - Secure software to remove cores during the boot up of a system.



## 2

strap - enables hardware to remove cores as GIC exits reset.

Type: `unsigned`

Default value: 0

### **redistributor-group**

Redistributor grouping information with affinity as JSON :

```
{
  "0": [
    "0.0.0.0",
    "0.0.0.1"
  ],
  "1": [
    "0.0.1.0",
    "0.0.1.1"
  ]
}
```

where RD with quad 0.0.0.0 and 0.0.0.1 belongs to RD group 0. All the RDs belong to one group when this parameter is not given.

Type: `string`

Default value: N/A

### **redistributor-group-file**

File path to redistributor grouping information with affinity as JSON.

The file uses the same format as `redistributor-group` parameter.

Type: `string`

Default value: N/A

### **redistributor-power-managed-by-pwrr**

GIC-720AE redistributor power management is done by updating GICR\_PWRR register.

Type: `bool`

Default value: `true`

### **reg-base**

GIC-720AE base address for memory-mapped register.

Type: `uint64_t`

Default value: `0x2c010000`

**reg-base-per-redistributor**

Base address for each redistributor in the form:

```
0.0.0.0=0x2c010000, 0.0.0.1=0x2c020000
```

All redistributors must be specified and this overrides the `reg-base` parameter (except that `reg-base` will still be used for the top-level redistributor). If `reg-base-per-redistributor-file` is specified, this parameter is ignored.

Type: `string`

Default value: `""`

**view-id-bits-offset**

The offset of the view-id bits in the address. Effective only when multiple views feature is enabled. Accepted values will depend on the configuration affecting the GIC memory footprint, such as the input to the parameter 'CPU-affinities'. For example, for a 32MB-sized view, the value of this parameter should be 25.

Type: `int`

Default value: 0

## 3.197 GIC720AE\_Filter

Defined in `LISA/GIC720AE.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
N/A	Preliminary support

For an explanation of the quality levels, see [Quality level definitions](#).

**About GIC720AE\_Filter**

The model has the same functionality as [GIC700\\_Filter](#), but in addition supports the following AE-specific features:

- GIC FMU
- Multiple views

It has the same limitations as [GIC700\\_Filter](#).

**Iris and MTI instances for GIC720AE\_Filter**

This model has the following Iris instances:

Name	Instance type
GIC720AE_Filter	GIC720AE
GIC720AE_Filter.GICV3_ProtocolChecker	GICv3ProtocolChecker
GIC720AE_Filter.ITS0	GICv4InterruptTranslationService
GIC720AE_Filter.ITS0.ExportTest.GIC720AE_Filter.ITS0.pvbus_m[0].pvbusmaster	PVBusMaster
GIC720AE_Filter.ITS0.bus_subordinate	PVBusSlave
GIC720AE_Filter.fmu	FMU
GIC720AE_Filter.fmu.pvbus_slave	PVBusSlave
GIC720AE_Filter.rd_0	GICv4RedistributorInternal
GIC720AE_Filter.rd_0_0	GICv4RedistributorInternal
GIC720AE_Filter.rd_0_0_0	GICv4RedistributorInternal
GIC720AE_Filter.rd_0_0_0_0	GICv3Redistributor
GIC720AE_Filter.rd_0_0_0_0.ExportTest.GIC720AE_Filter.rd_0_0_0_0.pvbus_m[0].pvbusmaster	PVBusMaster
GIC720AE_Filter.rd_tl	GICv3Distributor
GIC720AE_Filter.rd_tl.ExportTest.GIC720AE_Filter.rd_tl.pvbus_m[0].pvbusmaster	PVBusMaster

This model has the following MTI trace components:

Name	Component type
GIC720AE_Filter.GICV3_ProtocolChecker	GICv3ProtocolChecker
GIC720AE_Filter.ITS0	GICv4InterruptTranslationService
GIC720AE_Filter.ITS0.ExportTest.GIC720AE_Filter.ITS0.pvbus_m[0].pvbusmaster	PVBusMaster
GIC720AE_Filter.ITS0.bus_subordinate	PVBusSlave
GIC720AE_Filter.fmu	FMU
GIC720AE_Filter.fmu.pvbus_slave	PVBusSlave
GIC720AE_Filter.rd_0	GICv4RedistributorInternal
GIC720AE_Filter.rd_0_0	GICv4RedistributorInternal
GIC720AE_Filter.rd_0_0_0	GICv4RedistributorInternal
GIC720AE_Filter.rd_0_0_0_0	GICv3Redistributor
GIC720AE_Filter.rd_0_0_0_0.ExportTest.GIC720AE_Filter.rd_0_0_0_0.pvbus_m[0].pvbusmaster	PVBusMaster
GIC720AE_Filter.rd_tl	GICv3Distributor
GIC720AE_Filter.rd_tl.ExportTest.GIC720AE_Filter.rd_tl.pvbus_m[0].pvbusmaster	PVBusMaster

### Ports for GIC720AE\_Filter

Port	Direction	Protocol	Description
apb_bus	slave	PVBus	FMU signals
axi_stream_msi_s	slave	PVBus	AXI Stream Interface ports for MSI (message-signalled interrupt) translation, in GIC-720AE. Typically the SMMU's TCU connects to this port for MSI.

Port	Direction	Protocol	Description
chip_id	slave	Value	chip_id port for multichip operation
cpu_active_s	slave	Signal	CPUActive pins.
err_int	master	Signal	Error Recovery Interrupt
extended_ppi_in_0	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 0.
extended_ppi_in_100	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 100
extended_ppi_in_101	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 101
extended_ppi_in_102	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 102
extended_ppi_in_103	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 103
extended_ppi_in_104	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 104
extended_ppi_in_105	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 105
extended_ppi_in_106	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 106
extended_ppi_in_107	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 107
extended_ppi_in_108	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 108
extended_ppi_in_109	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 109
extended_ppi_in_10	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 10.
extended_ppi_in_110	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 110
extended_ppi_in_111	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 111
extended_ppi_in_112	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 112
extended_ppi_in_113	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 113
extended_ppi_in_114	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 114
extended_ppi_in_115	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 115
extended_ppi_in_116	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 116
extended_ppi_in_117	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 117
extended_ppi_in_118	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 118
extended_ppi_in_119	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 119
extended_ppi_in_11	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 11.
extended_ppi_in_120	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 120
extended_ppi_in_121	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 121
extended_ppi_in_122	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 122
extended_ppi_in_123	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 123
extended_ppi_in_124	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 124
extended_ppi_in_125	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 125
extended_ppi_in_126	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 126
extended_ppi_in_127	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 127
extended_ppi_in_128	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 128
extended_ppi_in_129	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 129
extended_ppi_in_12	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 12.
extended_ppi_in_130	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 130
extended_ppi_in_131	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 131
extended_ppi_in_132	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 132

Port	Direction	Protocol	Description
extended_ppi_in_133	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 133
extended_ppi_in_134	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 134
extended_ppi_in_135	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 135
extended_ppi_in_136	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 136
extended_ppi_in_137	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 137
extended_ppi_in_138	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 138
extended_ppi_in_139	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 139
extended_ppi_in_13	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 13.
extended_ppi_in_140	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 140
extended_ppi_in_141	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 141
extended_ppi_in_142	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 142
extended_ppi_in_143	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 143
extended_ppi_in_144	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 144
extended_ppi_in_145	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 145
extended_ppi_in_146	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 146
extended_ppi_in_147	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 147
extended_ppi_in_148	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 148
extended_ppi_in_149	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 149
extended_ppi_in_14	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 14.
extended_ppi_in_150	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 150
extended_ppi_in_151	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 151
extended_ppi_in_152	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 152
extended_ppi_in_153	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 153
extended_ppi_in_154	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 154
extended_ppi_in_155	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 155
extended_ppi_in_156	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 156
extended_ppi_in_157	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 157
extended_ppi_in_158	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 158
extended_ppi_in_159	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 159
extended_ppi_in_15	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 15.
extended_ppi_in_160	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 160
extended_ppi_in_161	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 161
extended_ppi_in_162	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 162
extended_ppi_in_163	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 163
extended_ppi_in_164	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 164
extended_ppi_in_165	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 165
extended_ppi_in_166	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 166
extended_ppi_in_167	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 167
extended_ppi_in_168	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 168
extended_ppi_in_169	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 169

Port	Direction	Protocol	Description
extended_ppi_in_16	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 16.
extended_ppi_in_170	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 170
extended_ppi_in_171	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 171
extended_ppi_in_172	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 172
extended_ppi_in_173	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 173
extended_ppi_in_174	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 174
extended_ppi_in_175	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 175
extended_ppi_in_176	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 176
extended_ppi_in_177	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 177
extended_ppi_in_178	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 178
extended_ppi_in_179	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 179
extended_ppi_in_17	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 17.
extended_ppi_in_180	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 180
extended_ppi_in_181	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 181
extended_ppi_in_182	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 182
extended_ppi_in_183	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 183
extended_ppi_in_184	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 184
extended_ppi_in_185	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 185
extended_ppi_in_186	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 186
extended_ppi_in_187	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 187
extended_ppi_in_188	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 188
extended_ppi_in_189	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 189
extended_ppi_in_18	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 18.
extended_ppi_in_190	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 190
extended_ppi_in_191	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 191
extended_ppi_in_192	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 192
extended_ppi_in_193	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 193
extended_ppi_in_194	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 194
extended_ppi_in_195	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 195
extended_ppi_in_196	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 196
extended_ppi_in_197	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 197
extended_ppi_in_198	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 198
extended_ppi_in_199	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 199
extended_ppi_in_19	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 19.
extended_ppi_in_1	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 1.
extended_ppi_in_200	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 200
extended_ppi_in_201	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 201
extended_ppi_in_202	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 202
extended_ppi_in_203	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 203
extended_ppi_in_204	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 204

Port	Direction	Protocol	Description
extended_ppi_in_205	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 205
extended_ppi_in_206	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 206
extended_ppi_in_207	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 207
extended_ppi_in_208	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 208
extended_ppi_in_209	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 209
extended_ppi_in_20	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 20.
extended_ppi_in_210	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 210
extended_ppi_in_211	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 211
extended_ppi_in_212	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 212
extended_ppi_in_213	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 213
extended_ppi_in_214	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 214
extended_ppi_in_215	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 215
extended_ppi_in_216	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 216
extended_ppi_in_217	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 217
extended_ppi_in_218	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 218
extended_ppi_in_219	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 219
extended_ppi_in_21	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 21.
extended_ppi_in_220	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 220
extended_ppi_in_221	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 221
extended_ppi_in_222	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 222
extended_ppi_in_223	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 223
extended_ppi_in_224	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 224
extended_ppi_in_225	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 225
extended_ppi_in_226	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 226
extended_ppi_in_227	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 227
extended_ppi_in_228	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 228
extended_ppi_in_229	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 229
extended_ppi_in_22	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 22.
extended_ppi_in_230	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 230
extended_ppi_in_231	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 231
extended_ppi_in_232	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 232
extended_ppi_in_233	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 233
extended_ppi_in_234	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 234
extended_ppi_in_235	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 235
extended_ppi_in_236	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 236
extended_ppi_in_237	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 237
extended_ppi_in_238	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 238
extended_ppi_in_239	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 239
extended_ppi_in_23	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 23.
extended_ppi_in_240	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 240



Port	Direction	Protocol	Description
extended_ppi_in_241	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 241
extended_ppi_in_242	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 242
extended_ppi_in_243	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 243
extended_ppi_in_244	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 244
extended_ppi_in_245	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 245
extended_ppi_in_246	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 246
extended_ppi_in_247	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 247
extended_ppi_in_248	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 248
extended_ppi_in_249	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 249
extended_ppi_in_24	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 24.
extended_ppi_in_250	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 250
extended_ppi_in_251	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 251
extended_ppi_in_252	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 252
extended_ppi_in_253	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 253
extended_ppi_in_254	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 254
extended_ppi_in_255	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 255
extended_ppi_in_25	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 25.
extended_ppi_in_26	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 26.
extended_ppi_in_27	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 27.
extended_ppi_in_28	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 28.
extended_ppi_in_29	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 29.
extended_ppi_in_2	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 2.
extended_ppi_in_30	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 30.
extended_ppi_in_31	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 31.
extended_ppi_in_32	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 32.
extended_ppi_in_33	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 33.
extended_ppi_in_34	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 34.
extended_ppi_in_35	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 35.
extended_ppi_in_36	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 36.
extended_ppi_in_37	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 37.
extended_ppi_in_38	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 38.
extended_ppi_in_39	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 39.
extended_ppi_in_3	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 3.
extended_ppi_in_40	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 40.
extended_ppi_in_41	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 41.
extended_ppi_in_42	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 42.
extended_ppi_in_43	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 43.
extended_ppi_in_44	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 44.
extended_ppi_in_45	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 45.
extended_ppi_in_46	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 46.



Port	Direction	Protocol	Description
extended_ppi_in_47	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 47.
extended_ppi_in_48	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 48.
extended_ppi_in_49	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 49.
extended_ppi_in_4	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 4.
extended_ppi_in_50	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 50.
extended_ppi_in_51	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 51.
extended_ppi_in_52	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 52.
extended_ppi_in_53	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 53.
extended_ppi_in_54	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 54.
extended_ppi_in_55	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 55.
extended_ppi_in_56	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 56.
extended_ppi_in_57	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 57.
extended_ppi_in_58	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 58.
extended_ppi_in_59	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 59.
extended_ppi_in_5	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 5.
extended_ppi_in_60	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 60.
extended_ppi_in_61	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 61.
extended_ppi_in_62	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 62.
extended_ppi_in_63	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 63.
extended_ppi_in_64	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 64.
extended_ppi_in_65	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 65.
extended_ppi_in_66	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 66.
extended_ppi_in_67	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 67.
extended_ppi_in_68	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 68.
extended_ppi_in_69	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 69.
extended_ppi_in_6	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 6.
extended_ppi_in_70	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 70.
extended_ppi_in_71	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 71.
extended_ppi_in_72	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 72.
extended_ppi_in_73	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 73.
extended_ppi_in_74	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 74.
extended_ppi_in_75	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 75.
extended_ppi_in_76	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 76.
extended_ppi_in_77	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 77.
extended_ppi_in_78	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 78.
extended_ppi_in_79	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 79.
extended_ppi_in_7	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 7.
extended_ppi_in_80	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 80.
extended_ppi_in_81	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 81.
extended_ppi_in_82	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 82.

Port	Direction	Protocol	Description
extended_ppi_in_83	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 83
extended_ppi_in_84	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 84
extended_ppi_in_85	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 85
extended_ppi_in_86	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 86
extended_ppi_in_87	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 87
extended_ppi_in_88	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 88
extended_ppi_in_89	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 89
extended_ppi_in_8	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 8.
extended_ppi_in_90	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 90
extended_ppi_in_91	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 91
extended_ppi_in_92	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 92
extended_ppi_in_93	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 93
extended_ppi_in_94	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 94
extended_ppi_in_95	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 95
extended_ppi_in_96	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 96
extended_ppi_in_97	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 97
extended_ppi_in_98	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 98
extended_ppi_in_99	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 99
extended_ppi_in_9	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 9.
extended_spi_in	slave	Signal	Extended Shared peripheral interrupts.
fault_int	master	Signal	RAS Interrupt signals Fault Handling Interrupt
fmu_cri	master	Signal	Critical Interrupt
fmu_eri	master	Signal	Error recovery Interrupt
icdrt_out	master	PVBus	AXI Stream Interface. This is used only when local cross chip addressing is enabled at the moment
icrdt_in	slave	PVBus	-
po_reset	slave	Signal	Reset.
ppi_in_0	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 0.
ppi_in_100	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 100
ppi_in_101	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 101
ppi_in_102	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 102
ppi_in_103	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 103
ppi_in_104	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 104
ppi_in_105	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 105
ppi_in_106	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 106
ppi_in_107	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 107
ppi_in_108	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 108
ppi_in_109	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 109
ppi_in_10	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 10.
ppi_in_110	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 110
ppi_in_111	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 111

Port	Direction	Protocol	Description
ppi_in_112	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 112
ppi_in_113	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 113
ppi_in_114	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 114
ppi_in_115	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 115
ppi_in_116	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 116
ppi_in_117	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 117
ppi_in_118	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 118
ppi_in_119	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 119
ppi_in_11	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 11.
ppi_in_120	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 120
ppi_in_121	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 121
ppi_in_122	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 122
ppi_in_123	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 123
ppi_in_124	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 124
ppi_in_125	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 125
ppi_in_126	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 126
ppi_in_127	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 127
ppi_in_128	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 128
ppi_in_129	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 129
ppi_in_12	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 12.
ppi_in_130	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 130
ppi_in_131	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 131
ppi_in_132	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 132
ppi_in_133	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 133
ppi_in_134	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 134
ppi_in_135	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 135
ppi_in_136	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 136
ppi_in_137	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 137
ppi_in_138	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 138
ppi_in_139	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 139
ppi_in_13	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 13.
ppi_in_140	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 140
ppi_in_141	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 141
ppi_in_142	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 142
ppi_in_143	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 143
ppi_in_144	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 144
ppi_in_145	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 145
ppi_in_146	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 146
ppi_in_147	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 147
ppi_in_148	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 148

Port	Direction	Protocol	Description
ppi_in_149	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 149
ppi_in_14	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 14.
ppi_in_150	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 150
ppi_in_151	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 151
ppi_in_152	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 152
ppi_in_153	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 153
ppi_in_154	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 154
ppi_in_155	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 155
ppi_in_156	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 156
ppi_in_157	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 157
ppi_in_158	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 158
ppi_in_159	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 159
ppi_in_15	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 15.
ppi_in_160	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 160
ppi_in_161	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 161
ppi_in_162	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 162
ppi_in_163	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 163
ppi_in_164	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 164
ppi_in_165	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 165
ppi_in_166	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 166
ppi_in_167	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 167
ppi_in_168	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 168
ppi_in_169	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 169
ppi_in_16	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 16.
ppi_in_170	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 170
ppi_in_171	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 171
ppi_in_172	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 172
ppi_in_173	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 173
ppi_in_174	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 174
ppi_in_175	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 175
ppi_in_176	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 176
ppi_in_177	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 177
ppi_in_178	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 178
ppi_in_179	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 179
ppi_in_17	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 17.
ppi_in_180	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 180
ppi_in_181	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 181
ppi_in_182	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 182
ppi_in_183	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 183
ppi_in_184	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 184

Port	Direction	Protocol	Description
ppi_in_185	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 185
ppi_in_186	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 186
ppi_in_187	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 187
ppi_in_188	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 188
ppi_in_189	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 189
ppi_in_18	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 18.
ppi_in_190	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 190
ppi_in_191	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 191
ppi_in_192	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 192
ppi_in_193	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 193
ppi_in_194	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 194
ppi_in_195	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 195
ppi_in_196	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 196
ppi_in_197	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 197
ppi_in_198	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 198
ppi_in_199	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 199
ppi_in_19	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 19.
ppi_in_1	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 1.
ppi_in_200	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 200
ppi_in_201	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 201
ppi_in_202	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 202
ppi_in_203	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 203
ppi_in_204	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 204
ppi_in_205	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 205
ppi_in_206	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 206
ppi_in_207	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 207
ppi_in_208	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 208
ppi_in_209	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 209
ppi_in_20	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 20.
ppi_in_210	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 210
ppi_in_211	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 211
ppi_in_212	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 212
ppi_in_213	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 213
ppi_in_214	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 214
ppi_in_215	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 215
ppi_in_216	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 216
ppi_in_217	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 217
ppi_in_218	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 218
ppi_in_219	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 219
ppi_in_21	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 21.

Port	Direction	Protocol	Description
ppi_in_220	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 220
ppi_in_221	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 221
ppi_in_222	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 222
ppi_in_223	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 223
ppi_in_224	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 224
ppi_in_225	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 225
ppi_in_226	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 226
ppi_in_227	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 227
ppi_in_228	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 228
ppi_in_229	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 229
ppi_in_22	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 22.
ppi_in_230	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 230
ppi_in_231	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 231
ppi_in_232	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 232
ppi_in_233	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 233
ppi_in_234	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 234
ppi_in_235	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 235
ppi_in_236	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 236
ppi_in_237	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 237
ppi_in_238	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 238
ppi_in_239	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 239
ppi_in_23	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 23.
ppi_in_240	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 240
ppi_in_241	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 241
ppi_in_242	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 242
ppi_in_243	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 243
ppi_in_244	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 244
ppi_in_245	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 245
ppi_in_246	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 246
ppi_in_247	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 247
ppi_in_248	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 248
ppi_in_249	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 249
ppi_in_24	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 24.
ppi_in_250	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 250
ppi_in_251	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 251
ppi_in_252	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 252
ppi_in_253	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 253
ppi_in_254	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 254
ppi_in_255	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 255
ppi_in_25	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 25.

Port	Direction	Protocol	Description
ppi_in_26	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 26.
ppi_in_27	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 27.
ppi_in_28	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 28.
ppi_in_29	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 29.
ppi_in_2	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 2.
ppi_in_30	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 30.
ppi_in_31	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 31.
ppi_in_32	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 32.
ppi_in_33	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 33.
ppi_in_34	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 34.
ppi_in_35	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 35.
ppi_in_36	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 36.
ppi_in_37	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 37.
ppi_in_38	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 38.
ppi_in_39	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 39.
ppi_in_3	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 3.
ppi_in_40	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 40.
ppi_in_41	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 41.
ppi_in_42	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 42.
ppi_in_43	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 43.
ppi_in_44	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 44.
ppi_in_45	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 45.
ppi_in_46	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 46.
ppi_in_47	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 47.
ppi_in_48	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 48.
ppi_in_49	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 49.
ppi_in_4	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 4.
ppi_in_50	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 50.
ppi_in_51	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 51.
ppi_in_52	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 52.
ppi_in_53	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 53.
ppi_in_54	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 54.
ppi_in_55	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 55.
ppi_in_56	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 56.
ppi_in_57	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 57.
ppi_in_58	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 58.
ppi_in_59	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 59.
ppi_in_5	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 5.
ppi_in_60	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 60.
ppi_in_61	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 61.



Port	Direction	Protocol	Description
ppi_in_62	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 62.
ppi_in_63	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 63.
ppi_in_64	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 64
ppi_in_65	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 65
ppi_in_66	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 66
ppi_in_67	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 67
ppi_in_68	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 68
ppi_in_69	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 69
ppi_in_6	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 6.
ppi_in_70	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 70
ppi_in_71	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 71
ppi_in_72	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 72
ppi_in_73	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 73
ppi_in_74	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 74
ppi_in_75	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 75
ppi_in_76	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 76
ppi_in_77	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 77
ppi_in_78	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 78
ppi_in_79	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 79
ppi_in_7	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 7.
ppi_in_80	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 80
ppi_in_81	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 81
ppi_in_82	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 82
ppi_in_83	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 83
ppi_in_84	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 84
ppi_in_85	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 85
ppi_in_86	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 86
ppi_in_87	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 87
ppi_in_88	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 88
ppi_in_89	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 89
ppi_in_8	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 8.
ppi_in_90	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 90
ppi_in_91	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 91
ppi_in_92	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 92
ppi_in_93	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 93
ppi_in_94	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 94
ppi_in_95	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 95
ppi_in_96	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 96
ppi_in_97	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 97
ppi_in_98	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 98



Port	Direction	Protocol	Description
ppi_in_99	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 99
ppi_in_9	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 9.
pvbus_filtermiss_m	master	PVBus	Memory bus out. Transactions not filtered by the component.
pvbus_m	master	PVBus	Memory bus out: transactions generated by the IRI.
pvbus_s	slave	PVBus	Memory bus in: memory-mapped register accesses are accepted on this interface.
redistributor_m	master	GICv3Comms	Input from and output to CPU interface.
reset	slave	Signal	Reset.
spi_in	slave	Signal	Shared peripheral interrupts.
wake_request	master	Signal	Power management outputs.

## Parameters for GIC720AE\_Filter

### CPU-affinities

A comma separated list of dotted quads containing the affinities of all PEs connected to this IRI. If CPU-affinities-file is specified, this parameter is ignored.

Type: string

Default value: N/A

### CPU-affinities-file

A file containing comma separated list of dotted quads containing the affinities of all PEs connected to this IRI. If this parameter is specified, CPU-affinities parameter will be ignored even when it is given.

Type: string

Default value: N/A

### GICD\_CTLR-DS-1-means-secure-only

If GICD\_CTLR.DS=1, GICD supports a single security state which is secure if this is true, otherwise is non-secure.

Type: bool

Default value: false

### GICD\_TYPER2

GICD\_TYPER2 value containing VID and VIL to define the width of vPEID for GICv4.1.

Type: uint32\_t

Default value: 0

**GICR-clear-enable-supported**

When true, this sets the value of the RO bit GICR\_CTLR.CES with the value of the parameter allow-LPIEN-clear, making it visible to software.

Type: `bool`

Default value: `false`

**GICR-invalidate-registers-implemented**

When true, the registers GICR\_INVLPIR, GICR\_INVALLR and GICR\_SYNCRR are implemented.

Type: `bool`

Default value: `false`

**IIDR**

GICD\_IIDR, GICR\_IIDR and GITS\_IIDR value. Defaults to the latest modelled revision.

Type: `uint32_t`

Default value: `0x0702143B`

**ITS-ID-bits**

Number of interrupt bits supported by ITS.

Type: `uint8_t`

Default value: `16`

**ITS-collection-ID-bits**

Number of collection bits supported by ITS (optional parameter, 0 => 16bits support and GITS\_TYPER.CIL=0).

Type: `uint8_t`

Default value: `8`

**ITS-count**

Number of Interrupt Translation Services to be instantiated (0=none).

Type: `uint8_t`

Default value: `1`

**ITS-device-bits**

Number of bits supported for ITS device IDs.

Type: `uint8_t`

Default value: 16

### **ITS-enable-itt-address-verification**

If true, a transaction will be sent to ITT Address for verification.

Type: `bool`

Default value: `false`

### **ITS-hardware-collection-count**

Number of hardware collections held exclusively in the ITS.

Type: `uint8_t`

Default value: 0

### **ITS-shared-vPE-table**

Number of affinity levels to which the vPE configuration table is shared. This parameter is valid when `has-gicv4.1` is true.

Type: `uint8_t`

Default value: 0

### **ITS-vmovp-bit**

Device supports software issuing a VMOVP to only one of the ITSs that has a mapping for a vPE. The device itself ensures synchronization of the VMOVP command across all ITSs that have mapping for that vPE.

Type: `bool`

Default value: `false`

### **PPI-count**

Selects the number of PPI available for each PE

**8**

id22-27,29,30

**12**

id 20-31

**16**

id 16-31.

Type: `int`

Default value: 16

**SPI-blocks**

Number of SPI blocks supported by the IRI, each block contains 32 SPIs.

Type: `int`

Default value: 62

**add-output-cpu-wake-request-signal-from-redistributor**

If true, the redistributor will have the output signal `cpu_wake_request` from GIC to DSU and if false, the signals are not added to the redistributor.

Type: `bool`

Default value: `false`

**affinity-width**

A dotted quad indicating the bitwidth of fields at each affinity level.

Type: `string`

Default value: "4.8.8.8"

**allow-LPIEN-clear**

Allow RW behaviour on `GICR_CTLR.LPIEN` instead of set once.

Type: `bool`

Default value: `true`

**chip-count**

The total number of chips supported.

Type: `int`

Default value: 16

**chip-id**

Chip ID for multichip operation.

Type: `uint8_t`

Default value: 0

**chip-select-affinity-level**

Affinity level 2 or 3 can be used for chip select.

Type: `int`

Default value: `3`

### **clear-ISPENDR-bit-for-level-sensitive-interrupt-when-acknowledged**

If true, acknowledgement of a level sensitive interrupt clears the corresponding bit in the ISPENDR register.

Type: `bool`

Default value: `false`

### **common-lpi-configuration**

Describes which re-distributors share (and must be configured with the same) LPI configuration table as described in GICR\_TYPER( 0:All, 1:A.x.x.x, 2:A.B.x.x, 3:A.B.C.x).

Type: `int`

Default value: `0`

### **common-vPE-table-affinity**

Affinity value list in the form of 'a.b.c.d, e.f.g.h, etc' given to the ITS(s), where a.b.c.d corresponds to ITS0, e.f.g.h corresponds to ITS1 and so on. Under an affinity value the vPE configuration table is shared among redistributors where the level to be shared is defined by ITS-shared-vPE-table. This parameter is valid when has-gicv4.1 is true.

Type: `string`

Default value: `N/A`

### **consolidators**

Specify consolidators' base addresses, interrupt line counts and base interrupt IDs, in the form:

```
'baseAddr0:itlineCount0:baseINTID0, baseAddr1:itlineCount1:baseINTID1'
```

For example:

```
'0x3f100000:64:4096, 0x3f200000:64:4224'
```

The consolidators' count is inferred from the list (maximum of 4). If not specified, the component contains no consolidators.

Type: `string`

Default value: `""`

**cross-chip-AMBA-is-ACE**

Indicates the AMBA protocol that the cross-chip interface uses. If true, the cross-chip interface uses the ACE5-Lite protocol. Otherwise, it uses the AXI5-Stream protocol. This parameter is meaningful only if enable-multichip-operation is set.

Type: `bool`

Default value: `false`

**enable-local-cross-chip-addressing**

If true, each distributor in a multichip system will use routing table address (ADDR) values programmed by local software, and will not be overwritten by the default chip. Otherwise, all the distributors will share the same routing table address values programmed in the default chip. This parameter is valid only if enable-multichip-operation is set.

Type: `bool`

Default value: `false`

**enable-multiple-views-feature**

If true, multiple view feature will provide multiple programming views which can be used by multiple hypervisors.

Type: `bool`

Default value: `false`

**enabled**

Enable GICv3 functionality; when false the component is inactive.

Type: `bool`

Default value: `true`

**extended-ppi-count**

Number of extended PPI supported.

Type: `unsigned`

Default value: `64`

**fm-blktype-num**

Number of stakeholder block types for FMU.

Type: `int`

Default value: `1`

**gicp-allow-ns-reset**

If true, non-secure read/write access to GICP register is allowed at reset. Not allowed otherwise. This emulates gicp\_allow\_ns tie-off signal.

Type: `bool`

Default value: `true`

**gict-allow-ns-reset**

If true, non-secure read/write access to GICT register is allowed at reset. Not allowed otherwise. This emulates gict\_allow\_ns tie-off signal.

Type: `bool`

Default value: `true`

**has-gicv4.1**

Enable GICv4.1 functionality; when false the component is inactive.

Type: `bool`

Default value: `true`

**has-two-security-states**

If true, has two security states.

Type: `bool`

Default value: `true`

**has\_VPENDBASER-dirty-flag-on-load**

GICR\_VPENDBASER.Dirty reflects transient loading state when valid=1.

Type: `bool`

Default value: `false`

**has\_nmi**

Enable support for Non-maskable Interrupts (NMIs).

Type: `bool`

Default value: `false`

**max-cores-supported-by-GCI**

GCI can support 1-64 cores. Maximum supported cores in GCI is configurable by device.

Type: `int`

Default value: 64

### **max-pe-on-chip**

Maximum number of cores on any single chip. This will be used to identify the target chip and core when multichip operation is enabled.

Type: `int`

Default value: 4

### **multichip-threaded-dgi**

Enable sending of multichip DGI messages in a separate thread, when multichip operation is enabled.

Type: `bool`

Default value: `true`

### **output\_attributes**

User-defined transform to be applied to bus attributes like ManagerID, ExtendedID or UserFlags. Currently, only works for MPAM Attributes encoding into bus attributes.

Type: `string`

Default value: "ExtendedID[62:55]=MPAM\_PMG, ExtendedID[54:39]=MPAM\_PARTID, ExtendedID[38]=MPAM\_NS"

### **print-memory-map**

Print memory map to stdout.

Type: `bool`

Default value: `false`

### **prog-mpidr**

Whether software or hardware can remove cores from a GIC configuration.

**0**

none

**1**

prog - Secure software to remove cores during the boot up of a system.

**2**

strap - enables hardware to remove cores as GIC exits reset.

Type: `unsigned`



Default value: 0

### **redistributor-group**

Redistributor grouping information with affinity as JSON :

```
{
  "0": [
    "0.0.0.0",
    "0.0.0.1"
  ],
  "1": [
    "0.0.1.0",
    "0.0.1.1"
  ]
}
```

where RD with quad 0.0.0.0 and 0.0.0.1 belongs to RD group 0. All the RDs belong to one group when this parameter is not given.

Type: `string`

Default value: N/A

### **redistributor-group-file**

File path to redistributor grouping information with affinity as JSON.

The file uses the same format as `redistributor-group` parameter.

Type: `string`

Default value: N/A

### **redistributor-power-managed-by-pwrr**

GIC-720AE redistributor power management is done by updating GICR\_PWRR register.

Type: `bool`

Default value: `true`

### **reg-base**

GIC-720AE base address for memory-mapped register.

Type: `uint64_t`

Default value: `0x2c010000`

### **reg-base-per-redistributor**

Base address for each redistributor in the form:

```
0.0.0.0=0x2c010000, 0.0.0.1=0x2c020000
```

All redistributors must be specified and this overrides the `reg-base` parameter (except that `reg-base` will still be used for the top-level redistributor). If `reg-base-per-redistributor-file` is specified, this parameter is ignored.

Type: `string`

Default value: `""`

#### **view-id-bits-offset**

The offset of the view-id bits in the address. Effective only when multiple views feature is enabled. Accepted values will depend on the configuration affecting the GIC memory footprint, such as the input to the parameter 'CPU-affinities'. For example, for a 32MB-sized view, the value of this parameter should be 25.

Type: `int`

Default value: 0

## 3.198 GIC\_400

Defined in `LISA/GIC_400.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p1	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### About GIC\_400

This component is a wrapper that permits easier configuration of the `v7_VGIC` component that supports parameterized configuration.

The GIC-400 has several memory-mapped interfaces at the same address. The processor that is communicating with the GIC-400 banks them. The GIC-400 must be able to identify which processor a transaction originates from. In the hardware, the `AUSER` fields on `AXI` supply this information to the GIC-400. In the model, there is no exact equivalent to this field. However, each transaction has a `manager_id` that the model can use to identify the originating processor.

Arm clusters assign the `manager_id` as follows:

#### **Bits[31:16]**

SBZ, which the GIC-400 ignores.

#### **Bits[5:2]**

CLUSTERID.

**Bits[1:0]**

`cpu_id` within the cluster.

CLUSTERID is the 4-bit field that either a parameter on the processor sets or a value that the `clusterid` port drives. CPUID is the core number within the cluster. CLUSTERID appears in the CP15 register space as part of the MPIDR register.

The Arm architecture suggests that each cluster in the system is given a different CLUSTERID. This distinction is essential for the VGIC to identify the cluster. The parameters in the GIC-400 component permit it to construct the map of `manager_id` to interface number.

Processor interfaces that the GIC-400 supports have these parameters:

- `interfaceN.cluster_id`
- `interfaceN.core_id`
- `interfaceN.inout_port_number_to_use`

N is the interface number (0-7). The `cluster_id` and `core_id` tell the GIC-400 to map that cluster or core combination to interface N.

By using `inout_port_number_to_use`, the GIC-400 has some input and output ports that pair with a particular processor interface. For example:

- The `irqcpu[]` pin wires to the `irq` port of the corresponding processor.
- The `cntpnsirq` pin from the processor wires to a `cntpnsirq[]` pin on GIC-400 to transport a Private Peripheral Interrupt (PPI) from the processor to the GIC-400.

The `interfaceN.inout_port_number_to_use` parameter supports clusters that can have variable numbers of cores. It tells the GIC-400 that to send to or receive a signal from the processor that is attached to interface N, it must use these pins:

- `irqout[interfaceN.inout_port_number_to_use]`
- `fiqout[interfaceN.inout_port_number_to_use]`
- `virgout[interfaceN.inout_port_number_to_use]`
- `vfiqout[interfaceN.inout_port_number_to_use]`
- `legacyirq[interfaceN.inout_port_number_to_use]`
- `cntpnsirq[interfaceN.inout_port_number_to_use]`
- `cntpsirq[interfaceN.inout_port_number_to_use]`
- `legacyfiq[interfaceN.inout_port_number_to_use]`
- `cntvirq[interfaceN.inout_port_number_to_use]`
- `cnthpirq[interfaceN.inout_port_number_to_use]`
- ...

`legacyirq` and `legacyfiq` are not signals from the processor but are signals into the GIC-400 from the legacy interrupt system. They are wired to PPIs. If the control registers of the GIC-400 are set

up in particular ways, they can also bypass the GIC-400. See [ARM Generic Interrupt Controller Architecture version 2.0 Architecture Specification](#) for more information.

The fabric between the clusters and the GIC might remap the `manager_id` of a transaction. If so, then the GIC might lose the ability to identify the originating processor. The fabrics that Arm ships in Fast Models perform no such transformation.

The comparison that the GIC-400 performs on the `manager_id` is only on the bottom 6 bits of the `manager_id`. It ignores the rest. If you are writing your own fabric and do not properly propagate the `manager_id` or transform it, the GIC-400 might not be able to identify the processor. The source code for the GIC\_400 component can be examined to see how it might be adapted for it to understand different `manager_id` schemes.

### Differences between the model and the RTL

The GIC-400 model has these limitations:

- Reads and writes to GICD\_ISACTIVER<sub>n</sub>, GICD\_ICACTIVER<sub>n</sub>, GICD\_ISPENDR<sub>n</sub>, or GICD\_ICPENDR<sub>n</sub> might not work as expected unless there is a configured target in GICD\_ICFGR<sub>m</sub>.
- Some of the interaction between GICD\_CTLR.EnableGrpX and level-sensitive interrupts might not work correctly.
- It does not model the nIRQOUT or nFIQOUT signals.
- It models interrupts with positive logic, rather than the negative logic that the hardware uses. So, the signal pins omit the 'n' prefix in their names.

### Iris and MTI instances for GIC\_400

This model has the following Iris instances:

Name	Instance type
GIC_400	GIC_400
GIC_400.vgic_bus_slave	PVBusSlave

This model has the following MTI trace components:

Name	Component type
GIC_400	GIC_400
GIC_400.vgic_bus_slave	PVBusSlave

### Ports for GIC\_400

Port	Direction	Protocol	Description
cfgsdisable	slave	Signal	Disable write access to some GIC registers.
cnthpirq	slave	Signal	Secure physical timer event. PPI interrupt id 26.
cntpnsirq	slave	Signal	Non-secure physical timer event. PPI interrupt id 30.
cntpsirq	slave	Signal	Secure physical timer event. PPI interrupt id 29.
cntvirq	slave	Signal	Virtual timer event. PPI interrupt id 27.

Port	Direction	Protocol	Description
fiqcpu	master	Signal	FIQ signal to the corresponding processor.
fiqout	master	Signal	FIQOUT signal to the corresponding processor.
irqcpu	master	Signal	IRQ signal to the corresponding processor.
irqout	master	Signal	IRQOUT signal to the corresponding processor.
irqs	slave	Signal	Interrupt request input lines for the GIC.
legacyfiq	slave	Signal	Signal into the GIC-400 from the legacy interrupt system. PPI interrupt id 28.
legacyirq	slave	Signal	Signal into the GIC-400 from the legacy interrupt system. PPI interrupt id 31.
pvbus_s	slave	PVBus	Handles incoming transactions from PVBus managers.
reset_signal	slave	Signal	Reset signal input.
vfiqcpu	master	Signal	Virtual FIQ signal to the processor.
virqcpu	master	Signal	Virtual IRQ signal to the processor.

## Parameters for GIC\_400

### NUM\_CPUS

number of interfaces to support.

Type: `uint32_t`

Default value: 1

### NUM\_SPIS

number of interrupt pins.

Type: `uint32_t`

Default value: 224

### enable\_log\_errors

Enable logging of errors.

Type: `bool`

Default value: `false`

### enable\_log\_fatal

Enable logging of fatal errors.

Type: `bool`

Default value: `false`

### enable\_log\_warnings

Enable logging of warnings.

Type: `bool`

Default value: `false`

### **`interface0.cluster_id`**

The CLUSTERID of the interface you want to appear as interface0 in the VGIC.

Type: `uint32_t`

Default value: 0

### **`interface0.core_id`**

The core id of interface0 in the cluster.

Type: `uint32_t`

Default value: 0

### **`interface0.inout_port_number_to_use`**

Which ppiN port is used for this interface.

Type: `uint32_t`

Default value: 0

### **`interface1.cluster_id`**

The CLUSTERID of the core you want to appear as interface1 in the VGIC.

Type: `uint32_t`

Default value: 0

### **`interface1.core_id`**

The core id of interface1 in the cluster.

Type: `uint32_t`

Default value: 0

### **`interface1.inout_port_number_to_use`**

Which ppiN port is used for this interface.

Type: `uint32_t`

Default value: 1

### **`interface2.cluster_id`**

The CLUSTERID of the interface you want to appear as 'core0' in the VGIC.

Type: `uint32_t`

Default value: 0

#### **`interface2.core_id`**

The core id of 'core0' in the cluster.

Type: `uint32_t`

Default value: 0

#### **`interface2.inout_port_number_to_use`**

Which ppiN port is used for this interface.

Type: `uint32_t`

Default value: 2

#### **`interface3.cluster_id`**

The CLUSTERID of the interface you want to appear as interface3 in the VGIC.

Type: `uint32_t`

Default value: 0

#### **`interface3.core_id`**

The core id of interface3 in the cluster.

Type: `uint32_t`

Default value: 0

#### **`interface3.inout_port_number_to_use`**

Which ppiN port is used for this interface.

Type: `uint32_t`

Default value: 3

#### **`interface4.cluster_id`**

The CLUSTERID of the interface you want to appear as interface4 in the VGIC.

Type: `uint32_t`

Default value: 0

#### **`interface4.core_id`**

The core id of interface4 in the cluster.

Type: uint32\_t

Default value: 0

#### **interface4.inout\_port\_number\_to\_use**

Which ppiN port is used for this interface.

Type: uint32\_t

Default value: 4

#### **interface5.cluster\_id**

The CLUSTERID of the interface you want to appear as interface5 in the VGIC.

Type: uint32\_t

Default value: 0

#### **interface5.core\_id**

The core id of interface5 in the cluster.

Type: uint32\_t

Default value: 0

#### **interface5.inout\_port\_number\_to\_use**

Which ppiN port is used for this interface.

Type: uint32\_t

Default value: 5

#### **interface6.cluster\_id**

The CLUSTERID of the interface you want to appear as interface6 in the VGIC.

Type: uint32\_t

Default value: 0

#### **interface6.core\_id**

The core id of interface6 in the cluster.

Type: uint32\_t

Default value: 0

#### **interface6.inout\_port\_number\_to\_use**

Which ppiN port is used for this interface.



Type: uint32\_t

Default value: 6

**interface7.cluster\_id**

The CLUSTERID of the interface you want to appear as interface7 in the VGIC.

Type: uint32\_t

Default value: 0

**interface7.core\_id**

The core id of interface7 in the cluster.

Type: uint32\_t

Default value: 0

**interface7.inout\_port\_number\_to\_use**

Which ppiN port is used for this interface.

Type: uint32\_t

Default value: 7

### 3.199 GIC\_IRI

Defined in LISA/GIC\_IRI.lisa.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

**About GIC\_IRI**

The GIC\_IRI has one slave PVBUS interface and one master PVBUS interface. It behaves in a similar manner to a GICv3-compatible device, with the slave interface, `pvbuss_s`, granting access to the register banks used by the configuration and operation of MSIs and the master interface, `pvbuss_m`, issuing transactions that are required by the ITS and the redistributors in LPI-related operations. All transactions that are routed to the slave port terminate in the component. Accesses to unmapped space are **RAZ/WI**.

An instance of GICv3 requires a small set of parameters to be configured to be useful. For example:

```
gic_iri : GIC_IRI(
    "reg-base" = 0xF0020000, //Base address for GICD_* REGISTERS, 64K aligned
    "CPU-affinities" = "0.0.0.0, 0.0.1.0, 0.0.1.1",
        //A comma-separated list of affinity addresses corresponding to
        //cpu affinities in the system
    "reg-base-per-redistributor"="0.0.0.0=0xF0040000,0.0.1.0=0xF0060000,
    0.0.0.0=0xF0080000",
        //Base addresses for each redistributor in a comma-separated list of
        //affinity=address
);
```

To use LPIs, an ITS must be configured. A minimal configuration might consist of, for example:

```
"ITS-count" = 1, //The number of ITSs in the IRI. Defaults to zero.
"ITS0-base" = 0xF0100000,
"GITS_BASER0-type" = 1, //Type 1 is Devices. A device table is always needed.
"GITS_BASER2-type" = 4, //Type 4 is Collections.
                        //A collection table is needed if GITS_TYPER.HCC is 0.
```

To use GICv4 functionality, one or more ITSs must be configured, as shown in the previous example. In addition, the following parameters are required:

```
"virtual-lpi-support"=true,
"GITS_BASER4-type"=2 //Type 2 is Virtual PEs.
                    //Such a table is needed for GICv4 functionality.
```



#### Note

- Set the `FASTSIM_GIC_MEMORY_MAP` environment variable to 1 to print to stderr the memory map of any GICv3 or later models that are included in the platform being run.
- Set the `GICD_ITARGETSR-RAZWI` parameter to true for legacy, GICv2-style routing, where interrupts are routed to the first processor in the system.

## Iris and MTI instances for GIC\_IRI

This model has the following Iris instances:

Name	Instance type
GIC_IRI	GIC_IRI
GIC_IRI.rd_0	GICv3RedistributorInternal
GIC_IRI.rd_0_0	GICv3RedistributorInternal
GIC_IRI.rd_0_0_0	GICv3RedistributorInternal
GIC_IRI.rd_0_0_0_0	GICv3Redistributor
GIC_IRI.rd_0_0_0_0.ExportTest.GIC_IRI.rd_0_0_0_0.pvbus_m[0].pvbusmaster	PVBusMaster
GIC_IRI.rd_t1	GICv3Distributor

This model has the following MTI trace components:

Name	Component type
GIC_IRI.rd_0	GLCv3RedistributorInternal
GIC_IRI.rd_0_0	GLCv3RedistributorInternal
GIC_IRI.rd_0_0_0	GLCv3RedistributorInternal
GIC_IRI.rd_0_0_0_0	GLCv3Redistributor
GIC_IRI.rd_0_0_0_0.ExportTest.GIC_IRI.rd_0_0_0_0.pvbus_m[0].pvbusmaster	PVBusMaster
GIC_IRI.rd_tl	GLCv3Distributor

## Ports for GIC\_IRI

Port	Direction	Protocol	Description
cfgsdisable	slave	Signal	Disable some SPLs signal.
extended_ppi_in_0	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 0.
extended_ppi_in_100	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 100.
extended_ppi_in_101	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 101.
extended_ppi_in_102	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 102.
extended_ppi_in_103	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 103.
extended_ppi_in_104	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 104.
extended_ppi_in_105	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 105.
extended_ppi_in_106	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 106.
extended_ppi_in_107	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 107.
extended_ppi_in_108	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 108.
extended_ppi_in_109	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 109.
extended_ppi_in_10	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 10.
extended_ppi_in_110	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 110.
extended_ppi_in_111	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 111.
extended_ppi_in_112	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 112.
extended_ppi_in_113	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 113.
extended_ppi_in_114	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 114.
extended_ppi_in_115	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 115.
extended_ppi_in_116	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 116.
extended_ppi_in_117	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 117.
extended_ppi_in_118	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 118.
extended_ppi_in_119	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 119.
extended_ppi_in_11	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 11.
extended_ppi_in_120	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 120.
extended_ppi_in_121	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 121.
extended_ppi_in_122	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 122.
extended_ppi_in_123	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 123.
extended_ppi_in_124	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 124.
extended_ppi_in_125	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 125.
extended_ppi_in_126	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 126.

Port	Direction	Protocol	Description
extended_ppi_in_127	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 127.
extended_ppi_in_128	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 128.
extended_ppi_in_129	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 129.
extended_ppi_in_12	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 12.
extended_ppi_in_130	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 130.
extended_ppi_in_131	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 131.
extended_ppi_in_132	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 132.
extended_ppi_in_133	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 133.
extended_ppi_in_134	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 134.
extended_ppi_in_135	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 135.
extended_ppi_in_136	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 136.
extended_ppi_in_137	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 137.
extended_ppi_in_138	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 138.
extended_ppi_in_139	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 139.
extended_ppi_in_13	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 13.
extended_ppi_in_140	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 140.
extended_ppi_in_141	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 141.
extended_ppi_in_142	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 142.
extended_ppi_in_143	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 143.
extended_ppi_in_144	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 144.
extended_ppi_in_145	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 145.
extended_ppi_in_146	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 146.
extended_ppi_in_147	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 147.
extended_ppi_in_148	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 148.
extended_ppi_in_149	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 149.
extended_ppi_in_14	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 14.
extended_ppi_in_150	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 150.
extended_ppi_in_151	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 151.
extended_ppi_in_152	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 152.
extended_ppi_in_153	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 153.
extended_ppi_in_154	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 154.
extended_ppi_in_155	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 155.
extended_ppi_in_156	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 156.
extended_ppi_in_157	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 157.
extended_ppi_in_158	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 158.
extended_ppi_in_159	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 159.
extended_ppi_in_15	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 15.
extended_ppi_in_160	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 160.
extended_ppi_in_161	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 161.
extended_ppi_in_162	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 162.

Port	Direction	Protocol	Description
extended_ppi_in_163	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 163.
extended_ppi_in_164	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 164.
extended_ppi_in_165	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 165.
extended_ppi_in_166	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 166.
extended_ppi_in_167	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 167.
extended_ppi_in_168	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 168.
extended_ppi_in_169	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 169.
extended_ppi_in_16	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 16.
extended_ppi_in_170	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 170.
extended_ppi_in_171	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 171.
extended_ppi_in_172	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 172.
extended_ppi_in_173	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 173.
extended_ppi_in_174	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 174.
extended_ppi_in_175	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 175.
extended_ppi_in_176	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 176.
extended_ppi_in_177	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 177.
extended_ppi_in_178	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 178.
extended_ppi_in_179	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 179.
extended_ppi_in_17	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 17.
extended_ppi_in_180	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 180.
extended_ppi_in_181	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 181.
extended_ppi_in_182	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 182.
extended_ppi_in_183	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 183.
extended_ppi_in_184	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 184.
extended_ppi_in_185	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 185.
extended_ppi_in_186	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 186.
extended_ppi_in_187	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 187.
extended_ppi_in_188	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 188.
extended_ppi_in_189	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 189.
extended_ppi_in_18	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 18.
extended_ppi_in_190	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 190.
extended_ppi_in_191	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 191.
extended_ppi_in_192	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 192.
extended_ppi_in_193	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 193.
extended_ppi_in_194	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 194.
extended_ppi_in_195	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 195.
extended_ppi_in_196	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 196.
extended_ppi_in_197	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 197.
extended_ppi_in_198	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 198.
extended_ppi_in_199	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 199.

Port	Direction	Protocol	Description
extended_ppi_in_19	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 19.
extended_ppi_in_1	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 1.
extended_ppi_in_200	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 200.
extended_ppi_in_201	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 201.
extended_ppi_in_202	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 202.
extended_ppi_in_203	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 203.
extended_ppi_in_204	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 204.
extended_ppi_in_205	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 205.
extended_ppi_in_206	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 206.
extended_ppi_in_207	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 207.
extended_ppi_in_208	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 208.
extended_ppi_in_209	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 209.
extended_ppi_in_20	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 20.
extended_ppi_in_210	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 210.
extended_ppi_in_211	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 211.
extended_ppi_in_212	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 212.
extended_ppi_in_213	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 213.
extended_ppi_in_214	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 214.
extended_ppi_in_215	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 215.
extended_ppi_in_216	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 216.
extended_ppi_in_217	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 217.
extended_ppi_in_218	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 218.
extended_ppi_in_219	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 219.
extended_ppi_in_21	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 21.
extended_ppi_in_220	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 220.
extended_ppi_in_221	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 221.
extended_ppi_in_222	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 222.
extended_ppi_in_223	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 223.
extended_ppi_in_224	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 224.
extended_ppi_in_225	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 225.
extended_ppi_in_226	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 226.
extended_ppi_in_227	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 227.
extended_ppi_in_228	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 228.
extended_ppi_in_229	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 229.
extended_ppi_in_22	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 22.
extended_ppi_in_230	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 230.
extended_ppi_in_231	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 231.
extended_ppi_in_232	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 232.
extended_ppi_in_233	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 233.
extended_ppi_in_234	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 234.

Port	Direction	Protocol	Description
extended_ppi_in_235	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 235.
extended_ppi_in_236	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 236.
extended_ppi_in_237	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 237.
extended_ppi_in_238	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 238.
extended_ppi_in_239	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 239.
extended_ppi_in_23	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 23.
extended_ppi_in_240	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 240.
extended_ppi_in_241	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 241.
extended_ppi_in_242	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 242.
extended_ppi_in_243	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 243.
extended_ppi_in_244	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 244.
extended_ppi_in_245	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 245.
extended_ppi_in_246	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 246.
extended_ppi_in_247	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 247.
extended_ppi_in_248	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 248.
extended_ppi_in_249	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 249.
extended_ppi_in_24	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 24.
extended_ppi_in_250	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 250.
extended_ppi_in_251	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 251.
extended_ppi_in_252	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 252.
extended_ppi_in_253	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 253.
extended_ppi_in_254	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 254.
extended_ppi_in_255	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 255.
extended_ppi_in_25	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 25.
extended_ppi_in_26	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 26.
extended_ppi_in_27	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 27.
extended_ppi_in_28	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 28.
extended_ppi_in_29	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 29.
extended_ppi_in_2	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 2.
extended_ppi_in_30	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 30.
extended_ppi_in_31	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 31.
extended_ppi_in_32	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 32.
extended_ppi_in_33	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 33.
extended_ppi_in_34	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 34.
extended_ppi_in_35	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 35.
extended_ppi_in_36	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 36.
extended_ppi_in_37	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 37.
extended_ppi_in_38	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 38.
extended_ppi_in_39	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 39.
extended_ppi_in_3	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 3.



Port	Direction	Protocol	Description
extended_ppi_in_40	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 40.
extended_ppi_in_41	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 41.
extended_ppi_in_42	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 42.
extended_ppi_in_43	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 43.
extended_ppi_in_44	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 44.
extended_ppi_in_45	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 45.
extended_ppi_in_46	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 46.
extended_ppi_in_47	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 47.
extended_ppi_in_48	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 48.
extended_ppi_in_49	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 49.
extended_ppi_in_4	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 4.
extended_ppi_in_50	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 50.
extended_ppi_in_51	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 51.
extended_ppi_in_52	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 52.
extended_ppi_in_53	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 53.
extended_ppi_in_54	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 54.
extended_ppi_in_55	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 55.
extended_ppi_in_56	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 56.
extended_ppi_in_57	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 57.
extended_ppi_in_58	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 58.
extended_ppi_in_59	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 59.
extended_ppi_in_5	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 5.
extended_ppi_in_60	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 60.
extended_ppi_in_61	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 61.
extended_ppi_in_62	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 62.
extended_ppi_in_63	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 63.
extended_ppi_in_64	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 64.
extended_ppi_in_65	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 65.
extended_ppi_in_66	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 66.
extended_ppi_in_67	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 67.
extended_ppi_in_68	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 68.
extended_ppi_in_69	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 69.
extended_ppi_in_6	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 6.
extended_ppi_in_70	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 70.
extended_ppi_in_71	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 71.
extended_ppi_in_72	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 72.
extended_ppi_in_73	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 73.
extended_ppi_in_74	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 74.
extended_ppi_in_75	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 75.
extended_ppi_in_76	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 76.



Port	Direction	Protocol	Description
extended_ppi_in_77	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 77.
extended_ppi_in_78	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 78.
extended_ppi_in_79	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 79.
extended_ppi_in_7	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 7.
extended_ppi_in_80	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 80.
extended_ppi_in_81	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 81.
extended_ppi_in_82	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 82.
extended_ppi_in_83	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 83.
extended_ppi_in_84	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 84.
extended_ppi_in_85	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 85.
extended_ppi_in_86	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 86.
extended_ppi_in_87	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 87.
extended_ppi_in_88	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 88.
extended_ppi_in_89	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 89.
extended_ppi_in_8	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 8.
extended_ppi_in_90	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 90.
extended_ppi_in_91	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 91.
extended_ppi_in_92	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 92.
extended_ppi_in_93	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 93.
extended_ppi_in_94	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 94.
extended_ppi_in_95	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 95.
extended_ppi_in_96	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 96.
extended_ppi_in_97	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 97.
extended_ppi_in_98	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 98.
extended_ppi_in_99	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 99.
extended_ppi_in_9	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 9.
extended_spi_in	slave	Signal	Extended Shared peripheral interrupts.
msi_error_interrupt	master	Signal	When report of MSI error allowed through interrupt, loop-back this signal to relevant IRQ input signal
po_reset	slave	Signal	Resets.
ppi_in_0	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 0.
ppi_in_100	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 100.
ppi_in_101	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 101.
ppi_in_102	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 102.
ppi_in_103	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 103.
ppi_in_104	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 104.
ppi_in_105	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 105.
ppi_in_106	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 106.
ppi_in_107	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 107.
ppi_in_108	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 108.
ppi_in_109	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 109.

Port	Direction	Protocol	Description
ppi_in_10	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 10.
ppi_in_110	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 110.
ppi_in_111	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 111.
ppi_in_112	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 112.
ppi_in_113	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 113.
ppi_in_114	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 114.
ppi_in_115	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 115.
ppi_in_116	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 116.
ppi_in_117	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 117.
ppi_in_118	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 118.
ppi_in_119	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 119.
ppi_in_11	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 11.
ppi_in_120	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 120.
ppi_in_121	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 121.
ppi_in_122	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 122.
ppi_in_123	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 123.
ppi_in_124	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 124.
ppi_in_125	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 125.
ppi_in_126	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 126.
ppi_in_127	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 127.
ppi_in_128	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 128.
ppi_in_129	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 129.
ppi_in_12	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 12.
ppi_in_130	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 130.
ppi_in_131	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 131.
ppi_in_132	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 132.
ppi_in_133	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 133.
ppi_in_134	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 134.
ppi_in_135	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 135.
ppi_in_136	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 136.
ppi_in_137	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 137.
ppi_in_138	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 138.
ppi_in_139	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 139.
ppi_in_13	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 13.
ppi_in_140	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 140.
ppi_in_141	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 141.
ppi_in_142	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 142.
ppi_in_143	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 143.
ppi_in_144	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 144.
ppi_in_145	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 145.

Port	Direction	Protocol	Description
ppi_in_146	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 146.
ppi_in_147	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 147.
ppi_in_148	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 148.
ppi_in_149	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 149.
ppi_in_14	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 14.
ppi_in_150	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 150.
ppi_in_151	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 151.
ppi_in_152	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 152.
ppi_in_153	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 153.
ppi_in_154	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 154.
ppi_in_155	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 155.
ppi_in_156	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 156.
ppi_in_157	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 157.
ppi_in_158	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 158.
ppi_in_159	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 159.
ppi_in_15	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 15.
ppi_in_160	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 160.
ppi_in_161	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 161.
ppi_in_162	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 162.
ppi_in_163	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 163.
ppi_in_164	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 164.
ppi_in_165	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 165.
ppi_in_166	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 166.
ppi_in_167	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 167.
ppi_in_168	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 168.
ppi_in_169	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 169.
ppi_in_16	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 16.
ppi_in_170	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 170.
ppi_in_171	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 171.
ppi_in_172	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 172.
ppi_in_173	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 173.
ppi_in_174	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 174.
ppi_in_175	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 175.
ppi_in_176	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 176.
ppi_in_177	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 177.
ppi_in_178	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 178.
ppi_in_179	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 179.
ppi_in_17	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 17.
ppi_in_180	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 180.
ppi_in_181	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 181.

Port	Direction	Protocol	Description
ppi_in_182	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 182.
ppi_in_183	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 183.
ppi_in_184	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 184.
ppi_in_185	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 185.
ppi_in_186	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 186.
ppi_in_187	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 187.
ppi_in_188	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 188.
ppi_in_189	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 189.
ppi_in_18	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 18.
ppi_in_190	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 190.
ppi_in_191	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 191.
ppi_in_192	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 192.
ppi_in_193	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 193.
ppi_in_194	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 194.
ppi_in_195	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 195.
ppi_in_196	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 196.
ppi_in_197	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 197.
ppi_in_198	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 198.
ppi_in_199	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 199.
ppi_in_19	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 19.
ppi_in_1	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 1.
ppi_in_200	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 200.
ppi_in_201	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 201.
ppi_in_202	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 202.
ppi_in_203	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 203.
ppi_in_204	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 204.
ppi_in_205	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 205.
ppi_in_206	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 206.
ppi_in_207	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 207.
ppi_in_208	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 208.
ppi_in_209	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 209.
ppi_in_20	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 20.
ppi_in_210	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 210.
ppi_in_211	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 211.
ppi_in_212	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 212.
ppi_in_213	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 213.
ppi_in_214	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 214.
ppi_in_215	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 215.
ppi_in_216	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 216.
ppi_in_217	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 217.

Port	Direction	Protocol	Description
ppi_in_218	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 218.
ppi_in_219	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 219.
ppi_in_21	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 21.
ppi_in_220	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 220.
ppi_in_221	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 221.
ppi_in_222	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 222.
ppi_in_223	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 223.
ppi_in_224	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 224.
ppi_in_225	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 225.
ppi_in_226	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 226.
ppi_in_227	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 227.
ppi_in_228	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 228.
ppi_in_229	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 229.
ppi_in_22	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 22.
ppi_in_230	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 230.
ppi_in_231	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 231.
ppi_in_232	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 232.
ppi_in_233	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 233.
ppi_in_234	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 234.
ppi_in_235	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 235.
ppi_in_236	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 236.
ppi_in_237	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 237.
ppi_in_238	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 238.
ppi_in_239	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 239.
ppi_in_23	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 23.
ppi_in_240	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 240.
ppi_in_241	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 241.
ppi_in_242	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 242.
ppi_in_243	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 243.
ppi_in_244	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 244.
ppi_in_245	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 245.
ppi_in_246	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 246.
ppi_in_247	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 247.
ppi_in_248	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 248.
ppi_in_249	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 249.
ppi_in_24	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 24.
ppi_in_250	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 250.
ppi_in_251	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 251.
ppi_in_252	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 252.
ppi_in_253	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 253.

Port	Direction	Protocol	Description
ppi_in_254	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 254.
ppi_in_255	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 255.
ppi_in_25	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 25.
ppi_in_26	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 26.
ppi_in_27	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 27.
ppi_in_28	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 28.
ppi_in_29	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 29.
ppi_in_2	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 2.
ppi_in_30	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 30.
ppi_in_31	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 31.
ppi_in_32	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 32.
ppi_in_33	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 33.
ppi_in_34	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 34.
ppi_in_35	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 35.
ppi_in_36	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 36.
ppi_in_37	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 37.
ppi_in_38	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 38.
ppi_in_39	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 39.
ppi_in_3	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 3.
ppi_in_40	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 40.
ppi_in_41	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 41.
ppi_in_42	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 42.
ppi_in_43	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 43.
ppi_in_44	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 44.
ppi_in_45	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 45.
ppi_in_46	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 46.
ppi_in_47	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 47.
ppi_in_48	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 48.
ppi_in_49	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 49.
ppi_in_4	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 4.
ppi_in_50	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 50.
ppi_in_51	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 51.
ppi_in_52	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 52.
ppi_in_53	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 53.
ppi_in_54	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 54.
ppi_in_55	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 55.
ppi_in_56	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 56.
ppi_in_57	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 57.
ppi_in_58	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 58.
ppi_in_59	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 59.

Port	Direction	Protocol	Description
ppi_in_5	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 5.
ppi_in_60	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 60.
ppi_in_61	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 61.
ppi_in_62	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 62.
ppi_in_63	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 63.
ppi_in_64	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 64.
ppi_in_65	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 65.
ppi_in_66	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 66.
ppi_in_67	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 67.
ppi_in_68	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 68.
ppi_in_69	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 69.
ppi_in_6	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 6.
ppi_in_70	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 70.
ppi_in_71	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 71.
ppi_in_72	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 72.
ppi_in_73	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 73.
ppi_in_74	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 74.
ppi_in_75	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 75.
ppi_in_76	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 76.
ppi_in_77	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 77.
ppi_in_78	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 78.
ppi_in_79	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 79.
ppi_in_7	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 7.
ppi_in_80	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 80.
ppi_in_81	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 81.
ppi_in_82	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 82.
ppi_in_83	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 83.
ppi_in_84	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 84.
ppi_in_85	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 85.
ppi_in_86	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 86.
ppi_in_87	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 87.
ppi_in_88	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 88.
ppi_in_89	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 89.
ppi_in_8	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 8.
ppi_in_90	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 90.
ppi_in_91	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 91.
ppi_in_92	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 92.
ppi_in_93	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 93.
ppi_in_94	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 94.
ppi_in_95	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 95.



Port	Direction	Protocol	Description
ppi_in_96	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 96.
ppi_in_97	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 97.
ppi_in_98	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 98.
ppi_in_99	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 99.
ppi_in_9	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 9.
pvbus_m	master	PVBus	Memory bus out: transactions generated by the IRI.
pvbus_s	slave	PVBus	Memory bus in: memory-mapped register accesses are accepted on this interface.
redistributor_m	master	GICv3Comms	Input from and output to CPU interface.
reset	slave	Signal	Resets.
spi_in	slave	Signal	Shared peripheral interrupts.
wake_request	master	Signal	Power management outputs.
wire_to_msi_in_0	slave	Signal	Wire-to-MSI interrupts for architectural consolidator 0.
wire_to_msi_in_1	slave	Signal	Wire-to-MSI interrupts for architectural consolidator 1.
wire_to_msi_in_2	slave	Signal	Wire-to-MSI interrupts for architectural consolidator 2.
wire_to_msi_in_3	slave	Signal	Wire-to-MSI interrupts for architectural consolidator 3.

## Parameters for GIC\_IRI

### A3-affinity-supported

Device supports affinity level 3 values that are non-zero.

Type: `bool`

Default value: `false`

### ARE-fixed-to-one

GICv2 compatibility is not supported and GICD\_CTLR.ARE\_\* is always one.

Type: `bool`

Default value: `false`

### CPU-affinities

A comma separated list of dotted quads containing the affinities of all PEs connected to this IRI. If CPU-affinities-file is specified, this parameter is ignored.

Type: `string`

Default value: `N/A`

### CPU-affinities-file

A file containing comma separated list of dotted quads containing the affinities of all PEs connected to this IRI. If this parameter is specified, CPU-affinities parameter will be ignored even when it is given.



Type: `string`

Default value: `N/A`

### **DPG-ARE-only**

Limit application of DPG bits to interrupt groups for which ARE=1.

Type: `bool`

Default value: `false`

### **DPG-bits-implemented**

Enable implementation of interrupt group participation bits or DPG bits in GICR\_CTLR.

Type: `bool`

Default value: `false`

### **DS-fixed-to-zero**

Enable/disable support of single security state.

Type: `bool`

Default value: `false`

### **GICD-alias**

In GICv2 mode: the base address for a 4k page alias of the first 4k of the Distributor page, in GICv3/GICv4 mode: the base address of a 64KB page containing message based SPI signalling register aliases(0:Disabled).

Type: `uint64_t`

Default value: `0x0`

### **GICD-legacy-registers-as-reserved**

When ARE is **RAO/WI**, makes superfluous registers in GICD reserved (including for the purpose of STATUSR updates).

Type: `bool`

Default value: `false`

### **GICD\_CTLR-DS-1-means-secure-only**

If GICD\_CTLR.DS=1, GICD supports a single security state which is secure if this is true, otherwise is non-secure.

Type: `bool`

Default value: `false`

**GICD\_ITARGETSR-RAZWI**

If true, the GICD\_ITARGETS registers are **RAZ/WI**.

Type: `bool`

Default value: `false`

**GICD\_PIDR**

The value for the GICD\_PIDR registers, if non-zero. Note: fixed fields (device type etc.) will be overridden in this value.

Type: `uint64_t`

Default value: `0x0`

**GICD\_TYPER2**

GICD\_TYPER2 value containing VID and VIL to define the width of vPEID for GICv4.1.

Type: `uint32_t`

Default value: `0`

**GICR-clear-enable-supported**

When true, this sets the value of the RO bit GICR\_CTLR.CES with the value of the parameter allow-LPIEN-clear, making it visible to software.

Type: `bool`

Default value: `false`

**GICR-invalidate-registers-implemented**

When true, the registers GICR\_INVLPIR, GICR\_INVALLR and GICR\_SYNCRR are implemented.

Type: `bool`

Default value: `false`

**GICR\_PIDR**

The value for the GICR\_PIDR registers, if non-zero. Note: fixed fields (device type etc.) will be overridden in this value.

Type: `uint64_t`

Default value: `0x0`

**GICR\_PROPBASER-read-only**

GICR\_PROPBASER register is read-only.

Type: `bool`

Default value: `false`

**GICR\_PROPBASER-reset-value**

Value of GICR\_PROPBASER on reset.

Type: `uint64_t`

Default value: `0x0`

**GITS\_BASER0-entry-bytes**

Number of bytes required per entry for GITS\_BASER0 register.

Type: `unsigned`

Default value: `8`

**GITS\_BASER0-indirect-RAZ**

Indirect field for GITS\_BASER0 register is **RAZ/WI**.

Type: `bool`

Default value: `false`

**GITS\_BASER0-type**

Type field for GITS\_BASER0 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: `uint8_t`

Default value: `0`

**GITS\_BASER1-entry-bytes**

Number of bytes required per entry for GITS\_BASER1 register.

Type: `unsigned`

Default value: `8`

**GITS\_BASER1-indirect-RAZ**

Indirect field for GITS\_BASER1 register is **RAZ/WI**.

Type: `bool`

Default value: `false`

**GITS\_BASER1-type**

Type field for GITS\_BASER1 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: `uint8_t`

Default value: 0

**GITS\_BASER2-entry-bytes**

Number of bytes required per entry for GITS\_BASER2 register.

Type: `unsigned`

Default value: 8

**GITS\_BASER2-indirect-RAZ**

Indirect field for GITS\_BASER2 register is **RAZ/WI**.

Type: `bool`

Default value: `false`

**GITS\_BASER2-type**

Type field for GITS\_BASER2 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: `uint8_t`

Default value: 0

**GITS\_BASER3-entry-bytes**

Number of bytes required per entry for GITS\_BASER3 register.

Type: `unsigned`

Default value: 8

**GITS\_BASER3-indirect-RAZ**

Indirect field for GITS\_BASER3 register is **RAZ/WI**.

Type: `bool`

Default value: `false`

**GITS\_BASER3-type**

Type field for GITS\_BASER3 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: `uint8_t`

Default value: 0

**GITS\_BASER4-entry-bytes**

Number of bytes required per entry for GITS\_BASER4 register.

Type: `unsigned`

Default value: 8

**GITS\_BASER4-indirect-RAZ**

Indirect field for GITS\_BASER4 register is **RAZ/WI**.

Type: `bool`

Default value: `false`

**GITS\_BASER4-type**

Type field for GITS\_BASER4 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: `uint8_t`

Default value: 0

**GITS\_BASER5-entry-bytes**

Number of bytes required per entry for GITS\_BASER5 register.

Type: `unsigned`

Default value: 8

**GITS\_BASER5-indirect-RAZ**

Indirect field for GITS\_BASER5 register is **RAZ/WI**.

Type: `bool`

Default value: `false`

**GITS\_BASER5-type**

Type field for GITS\_BASER5 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: `uint8_t`

Default value: 0

### **GITS\_BASER6-entry-bytes**

Number of bytes required per entry for GITS\_BASER6 register.

Type: `unsigned`

Default value: 8

### **GITS\_BASER6-indirect-RAZ**

Indirect field for GITS\_BASER6 register is **RAZ/WI**.

Type: `bool`

Default value: `false`

### **GITS\_BASER6-type**

Type field for GITS\_BASER6 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: `uint8_t`

Default value: 0

### **GITS\_BASER7-entry-bytes**

Number of bytes required per entry for GITS\_BASER7 register.

Type: `unsigned`

Default value: 8

### **GITS\_BASER7-indirect-RAZ**

Indirect field for GITS\_BASER7 register is **RAZ/WI**.

Type: `bool`

Default value: `false`

### **GITS\_BASER7-type**

Type field for GITS\_BASER7 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: `uint8_t`

Default value: 0

**GITS\_PIDR**

The value for the GITS\_PIDR registers, if non-zero. Note: fixed fields (device type etc.) will be overridden in this value.

Type: `uint64_t`

Default value: `0x0`

**ICFGR-PPI-mask**

Mask for writes to ICFGR registers that configure PPIs.

Type: `uint32_t`

Default value: `0xaaaaaaaa`

**ICFGR-PPI-reset**

Reset value for ICFGR registers that configure PPIs.

Type: `uint32_t`

Default value: `0x0`

**ICFGR-SGI-mask**

Mask for writes to ICFGR registers that configure SGIs.

Type: `uint32_t`

Default value: `0x0`

**ICFGR-SGI-reset**

Reset value for ICFGR registers that configure SGIs.

Type: `uint32_t`

Default value: `0xaaaaaaaa`

**ICFGR-SPI-mask**

Mask for writes to ICFGR registers that configure SPIs.

Type: `uint32_t`

Default value: `0xaaaaaaaa`

**ICFGR-SPI-reset**

Reset value for ICFGR registers that configure SPIs.

Type: `uint32_t`

Default value: `0x0`

### **ICFGR-rsvd-bit**

If ARE=0, the value of reserved bits i.e. bit 0,2,4..30 of ICFGRn for n>0.

Type: `bool`

Default value: `false`

### **IGROUP-PPI-mask**

Mask for writes to PPI bits in IGROUP registers.

Type: `uint16_t`

Default value: `0xffff`

### **IGROUP-PPI-reset**

Reset value for SGI bits in IGROUP registers.

Type: `uint16_t`

Default value: `0x0`

### **IGROUP-SGI-mask**

Mask for writes to SGI bits in IGROUP registers.

Type: `uint16_t`

Default value: `0xffff`

### **IGROUP-SGI-reset**

Reset value for SGI bits in IGROUP registers.

Type: `uint16_t`

Default value: `0x0`

### **IIDR**

GICD\_IIDR and GICR\_IIDR value.

Type: `uint32_t`

Default value: `0x0`

### **IRI-ID-bits**

Number of bits used to represent interrupts IDs in the Distributor and Redistributors, forced to 10 when none of LPIs, extended SPIs or extended PPIs is supported.



Type: `int`

Default value: `16`

#### **IROUTER-IRM-RAZ-WI**

GICD\_IROUTERn.InterruptRoutingMode is **RAZ/WI**.

Type: `bool`

Default value: `false`

#### **ITS-BASER-force-page-alignment**

Force alignment of address written to a GITS\_BASER register to the page size configured.

Type: `bool`

Default value: `true`

#### **ITS-ID-bits**

Number of interrupt bits supported by ITS.

Type: `uint8_t`

Default value: `16`

#### **ITS-MOVALL-update-collections**

Whether MOVALL command updates the collection entirely.

Type: `bool`

Default value: `false`

#### **ITS-TRANSLATE64R**

Add an implementation specific register at `0x10008` supporting 64 bit TRANSLATER (`dev[63:32]`, `interrupt[31:0]`).

Type: `bool`

Default value: `false`

#### **ITS-cache-invalidate-on-disable**

Sets the RO bit GITS\_TYPER.INV. When true, after the following sequence: 1) GITS\_CTLR.Enabled written 1->0, 2) GITS\_CTLR.Quiescent observed as 1, 3) GITS\_BASER<n>.Valid written 1->0, there is no cached information from the ITS memory structure pointed to by GITS\_BASER<n>.

Type: `bool`

Default value: `false`

**ITS-collection-ID-bits**

Number of collection bits supported by ITS (optional parameter, 0 => 16bits support and GITS\_TYPER.CIL=0).

Type: `int`

Default value: 0

**ITS-count**

Number of Interrupt Translation Services to be instantiated (0=none).

Type: `uint8_t`

Default value: 0

**ITS-cumulative-collection-tables**

When true, the supported amount of collections is the sum of GITS\_TYPER.HCC and the number of collections supported in memory, otherwise, simply the number supported in memory only. Irrelevant when HCC=0.

Type: `bool`

Default value: `true`

**ITS-device-bits**

Number of bits supported for ITS device IDs.

Type: `int`

Default value: 16

**ITS-enable-itt-address-verification**

If true, a transaction will be sent to ITT Address for verification.

Type: `bool`

Default value: `false`

**ITS-entry-size**

Number of bytes required to store each entry in the ITT tables.

Type: `int`

Default value: 8

**ITS-hardware-collection-count**

Number of hardware collections held exclusively in the ITS.

Type: `int`

Default value: `0`

### **ITS-legacy-iidr-typer-offset**

Put the GITS\_IIDR and GITS\_TYPER registers at their older offset of `0x8` and `0x4` respectively.

Type: `bool`

Default value: `false`

### **ITS-shared-vPE-table**

Number of affinity levels to which the vPE configuration table is shared. This parameter is valid when `has-gicv4.1` is true.

Type: `uint8_t`

Default value: `0`

### **ITS-threaded-command-queue**

Enable execution of ITS commands in a separate thread which is sometimes required for cosimulation.

Type: `bool`

Default value: `true`

### **ITS-use-physical-target-addresses**

Use physical hardware addresses for targets in ITS commands – must be true for distributed implementations.

Type: `bool`

Default value: `true`

### **ITS-vmovp-bit**

Device supports software issuing a VMOVP to only one of the ITSs that has a mapping for a vPE. The device itself ensures synchronization of the VMOVP command across all ITSs that have mapping for that vPE.

Type: `bool`

Default value: `false`

### **ITS0-base**

Register base address for ITS0 (automatic if 0).

Type: `uint64_t`

Default value: 0

**ITS1-base**

Register base address for ITS1 (automatic if 0).

Type: `uint64_t`

Default value: 0

**ITS2-base**

Register base address for ITS2 (automatic if 0).

Type: `uint64_t`

Default value: 0

**ITS3-base**

Register base address for ITS3 (automatic if 0).

Type: `uint64_t`

Default value: 0

**LPI-cache-check-data**

Enable Cached LPI data against memory checking when available for cache type.

Type: `bool`

Default value: `false`

**LPI-cache-type**

Cache type for LPIs, 0:No caching, 1:Full caching.

Type: `uint8_t`

Default value: 1

**MSI\_IIDR**

Value returned in MSI\_IIDR registers.

Type: `uint32_t`

Default value: 0x0

**MSI\_NS-frame0-base**

If non-zero, sets the base address used for non-secure MSI frame 0 registers.

Type: `uint64_t`

Default value: 0x0

**MSI\_NS-frame0-max-SPI**

Maximum SPI ID supported by non-secure MSI frame 0. Set to 0 to disable frame.

Type: uint16\_t

Default value: 0

**MSI\_NS-frame0-min-SPI**

Minimum SPI ID supported by non-secure MSI frame 0. Set to 0 to disable frame.

Type: uint16\_t

Default value: 0

**MSI\_NS-frame1-base**

If non-zero, sets the base address used for non-secure MSI frame 1 registers.

Type: uint64\_t

Default value: 0x0

**MSI\_NS-frame1-max-SPI**

Maximum SPI ID supported by non-secure MSI frame 1. Set to 0 to disable frame.

Type: uint16\_t

Default value: 0

**MSI\_NS-frame1-min-SPI**

Minimum SPI ID supported by non-secure MSI frame 1. Set to 0 to disable frame.

Type: uint16\_t

Default value: 0

**MSI\_NS-frame2-base**

If non-zero, sets the base address used for non-secure MSI frame 2 registers.

Type: uint64\_t

Default value: 0x0

**MSI\_NS-frame2-max-SPI**

Maximum SPI ID supported by non-secure MSI frame 2. Set to 0 to disable frame.

Type: uint16\_t

Default value: 0

**MSI\_NS-frame2-min-SPI**

Minimum SPI ID supported by non-secure MSI frame 2. Set to 0 to disable frame.

Type: uint16\_t

Default value: 0

**MSI\_NS-frame3-base**

If non-zero, sets the base address used for non-secure MSI frame 3 registers.

Type: uint64\_t

Default value: 0x0

**MSI\_NS-frame3-max-SPI**

Maximum SPI ID supported by non-secure MSI frame 3. Set to 0 to disable frame.

Type: uint16\_t

Default value: 0

**MSI\_NS-frame3-min-SPI**

Minimum SPI ID supported by non-secure MSI frame 3. Set to 0 to disable frame.

Type: uint16\_t

Default value: 0

**MSI\_NS-frame4-base**

If non-zero, sets the base address used for non-secure MSI frame 4 registers.

Type: uint64\_t

Default value: 0x0

**MSI\_NS-frame4-max-SPI**

Maximum SPI ID supported by non-secure MSI frame 4. Set to 0 to disable frame.

Type: uint16\_t

Default value: 0

**MSI\_NS-frame4-min-SPI**

Minimum SPI ID supported by non-secure MSI frame 4. Set to 0 to disable frame.

Type: uint16\_t

Default value: 0

**MSI\_NS-frame5-base**

If non-zero, sets the base address used for non-secure MSI frame 5 registers.

Type: uint64\_t

Default value: 0x0

**MSI\_NS-frame5-max-SPI**

Maximum SPI ID supported by non-secure MSI frame 5. Set to 0 to disable frame.

Type: uint16\_t

Default value: 0

**MSI\_NS-frame5-min-SPI**

Minimum SPI ID supported by non-secure MSI frame 5. Set to 0 to disable frame.

Type: uint16\_t

Default value: 0

**MSI\_NS-frame6-base**

If non-zero, sets the base address used for non-secure MSI frame 6 registers.

Type: uint64\_t

Default value: 0x0

**MSI\_NS-frame6-max-SPI**

Maximum SPI ID supported by non-secure MSI frame 6. Set to 0 to disable frame.

Type: uint16\_t

Default value: 0

**MSI\_NS-frame6-min-SPI**

Minimum SPI ID supported by non-secure MSI frame 6. Set to 0 to disable frame.

Type: uint16\_t

Default value: 0

**MSI\_NS-frame7-base**

If non-zero, sets the base address used for non-secure MSI frame 7 registers.

Type: uint64\_t

Default value: 0x0

**MSI\_NS-frame7-max-SPI**

Maximum SPI ID supported by non-secure MSI frame 7. Set to 0 to disable frame.

Type: uint16\_t

Default value: 0

**MSI\_NS-frame7-min-SPI**

Minimum SPI ID supported by non-secure MSI frame 7. Set to 0 to disable frame.

Type: uint16\_t

Default value: 0

**MSI\_PIDR**

The value for the MSI\_PIDR registers, if non-zero and distributor supports GICv2m. Note: fixed fields (device type etc.) will be overridden in this value.

Type: uint64\_t

Default value: 0x0

**MSI\_S-frame0-base**

If non-zero, sets the base address used for secure MSI frame 0 registers.

Type: uint64\_t

Default value: 0x0

**MSI\_S-frame0-max-SPI**

Maximum SPI ID supported by secure MSI frame 0. Set to 0 to disable frame.

Type: uint16\_t

Default value: 0

**MSI\_S-frame0-min-SPI**

Minimum SPI ID supported by secure MSI frame 0. Set to 0 to disable frame.

Type: uint16\_t

Default value: 0

**MSI\_S-frame1-base**

If non-zero, sets the base address used for secure MSI frame 1 registers.



Type: uint64\_t

Default value: 0x0

#### **MSI\_S-frame1-max-SPI**

Maximum SPI ID supported by secure MSI frame 1. Set to 0 to disable frame.

Type: uint16\_t

Default value: 0

#### **MSI\_S-frame1-min-SPI**

Minimum SPI ID supported by secure MSI frame 1. Set to 0 to disable frame.

Type: uint16\_t

Default value: 0

#### **MSI\_S-frame2-base**

If non-zero, sets the base address used for secure MSI frame 2 registers.

Type: uint64\_t

Default value: 0x0

#### **MSI\_S-frame2-max-SPI**

Maximum SPI ID supported by secure MSI frame 2. Set to 0 to disable frame.

Type: uint16\_t

Default value: 0

#### **MSI\_S-frame2-min-SPI**

Minimum SPI ID supported by secure MSI frame 2. Set to 0 to disable frame.

Type: uint16\_t

Default value: 0

#### **MSI\_S-frame3-base**

If non-zero, sets the base address used for secure MSI frame 3 registers.

Type: uint64\_t

Default value: 0x0

#### **MSI\_S-frame3-max-SPI**

Maximum SPI ID supported by secure MSI frame 3. Set to 0 to disable frame.

Type: uint16\_t

Default value: 0

#### **MSI\_S-frame3-min-SPI**

Minimum SPI ID supported by secure MSI frame 3. Set to 0 to disable frame.

Type: uint16\_t

Default value: 0

#### **MSI\_S-frame4-base**

If non-zero, sets the base address used for secure MSI frame 4 registers.

Type: uint64\_t

Default value: 0x0

#### **MSI\_S-frame4-max-SPI**

Maximum SPI ID supported by secure MSI frame 4. Set to 0 to disable frame.

Type: uint16\_t

Default value: 0

#### **MSI\_S-frame4-min-SPI**

Minimum SPI ID supported by secure MSI frame 4. Set to 0 to disable frame.

Type: uint16\_t

Default value: 0

#### **MSI\_S-frame5-base**

If non-zero, sets the base address used for secure MSI frame 5 registers.

Type: uint64\_t

Default value: 0x0

#### **MSI\_S-frame5-max-SPI**

Maximum SPI ID supported by secure MSI frame 5. Set to 0 to disable frame.

Type: uint16\_t

Default value: 0

#### **MSI\_S-frame5-min-SPI**

Minimum SPI ID supported by secure MSI frame 5. Set to 0 to disable frame.

Type: `uint16_t`

Default value: 0

### **MSI\_S-frame6-base**

If non-zero, sets the base address used for secure MSI frame 6 registers.

Type: `uint64_t`

Default value: 0x0

### **MSI\_S-frame6-max-SPI**

Maximum SPI ID supported by secure MSI frame 6. Set to 0 to disable frame.

Type: `uint16_t`

Default value: 0

### **MSI\_S-frame6-min-SPI**

Minimum SPI ID supported by secure MSI frame 6. Set to 0 to disable frame.

Type: `uint16_t`

Default value: 0

### **MSI\_S-frame7-base**

If non-zero, sets the base address used for secure MSI frame 7 registers.

Type: `uint64_t`

Default value: 0x0

### **MSI\_S-frame7-max-SPI**

Maximum SPI ID supported by secure MSI frame 7. Set to 0 to disable frame.

Type: `uint16_t`

Default value: 0

### **MSI\_S-frame7-min-SPI**

Minimum SPI ID supported by secure MSI frame 7. Set to 0 to disable frame.

Type: `uint16_t`

Default value: 0

### **PA\_SIZE**

Number of valid bits in physical address.

Type: `int`

Default value: 48

### **PPI-implemented-mask**

Mask of PPIs that are implemented. One bit per PPI bit 0 == PPI 16 (first PPI). This will affect other masks.

Type: `uint16_t`

Default value: `0xffff`

### **SPI-count**

Number of SPIs that are implemented.

Type: `uint16_t`

Default value: 224

### **SPI-message-based-support**

Distributor supports message based signaling of SPI.

Type: `bool`

Default value: `true`

### **SPI-unimplemented**

A comma separated list of unimplemented SPIs ranges for sparse SPI definition(for ex: '35, 39-42, 73').

Type: `string`

Default value: ""

### **STATUSR-implemented**

Determines whether the GICR\_STATUSR register is implemented.

Type: `bool`

Default value: `true`

### **add-output-cpu-wake-request-signal-from-redistributor**

If true, the redistributor will have the output signal `cpu_wake_request` from GIC to DSU and if false, the signals are not added to the redistributor.

Type: `bool`

Default value: `false`

**allow-LPIEN-clear**

Allow RW behaviour on GICR\_CTLR.LPIEN instead of set once.

Type: `bool`

Default value: `false`

**clear-ISPENDR-bit-for-level-sensitive-interrupt-when-acknowledged**

If true, acknowledgement of a level sensitive interrupt clears the corresponding bit in the ISPENDR register.

Type: `bool`

Default value: `false`

**common-lpi-configuration**

Describes which re-distributors share (and must be configured with the same) LPI configuration table as described in GICR\_TYPER( 0:All, 1:A.x.x.x, 2:A.B.x.x, 3:A.B.C.x).

Type: `int`

Default value: `0`

**common-vPE-table-affinity**

Affinity value list in the form of 'a.b.c.d, e.f.g.h, etc' given to the ITS(s) (where a.b.c.d corresponds to ITS0, e.f.g.h corresponds to ITS1 and so on). Under an affinity value the vPE configuration table is shared among redistributors where the level to be shared is defined by ITS-shared-vPE-table. This parameter is valid when has-gicv4.1 is true.

Type: `string`

Default value: `""`

**consolidators**

Specify consolidators' base addresses, interrupt line counts and base interrupt IDs, in the form 'baseAddr0:itlineCount0:baseINTID0, baseAddr1:itlineCount1:baseINTID1, [etc]' (eg '0x3f100000:64:4096, 0x3f200000:64:4224'). The consolidators' count is inferred from the list (maximum of 4). If not specified, the component contains no consolidators.

Type: `string`

Default value: `""`

**delay-ITS-accesses**

Delay accesses from the ITS until GICR\_SYNCR is read.

Type: `bool`

Default value: `true`

### **delay-redistributor-accesses**

Delay memory accesses from the redistributor until GICR\_SYNCR is read.

Type: `bool`

Default value: `true`

### **direct-lpi-support**

Enable support for LPI operations through GICR registers.

Type: `bool`

Default value: `false`

### **enable\_protocol\_checking**

Enable/disable protocol checking at cpu interface.

Type: `bool`

Default value: `false`

### **enabled**

Enable GICv3 functionality; when false the component is inactive.

Type: `bool`

Default value: `true`

### **extended-ppi-count**

Number of extended PPI supported.

Type: `unsigned`

Default value: `0`

### **extended-spi-count**

Number of extended SPI supported.

Type: `unsigned`

Default value: `0`

### **fixed-routed-spis**

Value of IROUTER[n] register in the form 'n=a.b.c.d, n='. *The RM bit of IROUTER is 0 when n=a.b.c.d is used else 1 when n= is used.* n can be  $\geq 32$  and  $\leq 1019$ .

Type: `string`

Default value: `""`

### **`gicr-icfgr-extended-count`**

Number of extended GICR\_ICFGR registers supported.

Type: `uint8_t`

Default value: `4`

### **`gicv2-only`**

If true, when using the GICv3/GICv4 model, pretend to be a GICv2 system.

Type: `bool`

Default value: `false`

### **`group-enables-control-doorbell`**

When true, GICR\_VPENDBASER.{VGrp0En,VGrp1En} are cached to allow GIC to check group enables when virtual interrupt targeting this VCPU which is non-resident reaches Redistributor.

Type: `bool`

Default value: `false`

### **`has-gicv4.1`**

Enable GICv4.1 functionality; when false the component is inactive.

Type: `bool`

Default value: `false`

### **`has-two-security-states`**

If true, has two security states.

Type: `bool`

Default value: `true`

### **`has_VPENDBASER-dirty-flag-on-load`**

GICR\_VPENDBASER.Dirty reflects transient loading state when valid=1.

Type: `bool`

Default value: `false`

**has\_mpam**

Implement ARMv8.4 MPAM Registers and associated functionality. values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: 0

**has\_nmi**

Enable support for Non-maskable Interrupts (NMIs).

Type: `bool`

Default value: `false`

**ignore-generate-sgi-when-no-are**

Ignore GenerateSGL packets coming from the CPU interface if both ARE\_S and ARE\_NS are 0.

Type: `bool`

Default value: `false`

**individual-doorbell-not-supported**

For IRI with support of virtual interrupt, individual doorbell is not supported when true.

Type: `bool`

Default value: `false`

**irouter-default-mask**

Default Mask value for IROUTER[32..1019] register in the form 'a.b.c.d'.

Type: `string`

Default value: ""

**irouter-default-reset**

Default Reset Value of IROUTER[32..1019] register in the form 'a.b.c.d' or \*.

Type: `string`

Default value: N/A

**irouter-mask-values**

Mask Value of IROUTER[n] register in the form 'n=a.b.c.d'. n can be  $\geq 32$  and  $\leq 1019$ .



Type: `string`

Default value: `""`

### **irouter-reset-values**

Reset Value of IROUTER[n] register in the form 'n=a.b.c.d or n=\*.n can be  $\geq 32$  and  $\leq 1019$ .

Type: `string`

Default value: `N/A`

### **legacy-sgi-enable-rao**

Enables for SGI associated with an ARE=0 regime are **RAO/WI**.

Type: `bool`

Default value: `false`

### **local-SEIs**

Generate SEI to signal internal issues.

Type: `bool`

Default value: `false`

### **local-VSEIs**

Generate VSEI to signal internal issues.

Type: `bool`

Default value: `false`

### **lockable-SPI-count**

Number of SPIs that are locked down when CFGSDISABLE signal is asserted. Only applies for GICv2.

Type: `uint8_t`

Default value: `0`

### **monolithic**

Indicate that the implementation is not distributed.

Type: `bool`

Default value: `false`

**mpam\_max\_partid**

MPAM Maximum PARTID Supported.

Type: `uint16_t`

Default value: `0xffff`

**mpam\_max\_pmg**

MPAM Maximum PMG Supported.

Type: `uint8_t`

Default value: `255`

**non-ARE-core-count**

Maximum number of non-ARE cores; normally used to pass the cluster-level `NUM_CORES` parameter to the top-level redistributor.

Type: `int`

Default value: `8`

**outer-cacheability-support**

Allow configuration of outer cacheability attributes in ITS and Redistributor.

Type: `bool`

Default value: `false`

**output\_attributes**

User-defined transform to be applied to bus attributes like `ManagerID`, `ExtendedID` or `UserFlags`. Currently, only works for MPAM Attributes encoding into bus attributes.

Type: `string`

Default value: `"ExtendedID[62:55]=MPAM_PMG, ExtendedID[54:39]=MPAM_PARTID, ExtendedID[38]=MPAM_NS"`

**print-memory-map**

Print memory map to stdout.

Type: `bool`

Default value: `false`

**priority-bits**

Number of implemented priority bits.

Type: `uint8_t`

Default value: 5

### **processor-numbers**

Specify processor numbers (as appears in GICR\_TYPER) in the form 0.0.0.0=0,0.0.0.1=1 etc.) If not specified, will number processors starting at 0.

Type: `string`

Default value: ""

### **redistributor-threaded-sync**

Enable execution of redistributor delayed transactions in a separate thread which is sometimes required for cosimulation.

Type: `bool`

Default value: `true`

### **reg-base**

Base for decoding GICv3/GICv4 registers.

Type: `uint64_t`

Default value: `0x2c010000`

### **reg-base-per-redistributor**

Base address for each redistributor in the form:

```
0.0.0.0=0x2c010000, 0.0.0.1=0x2c020000
```

All redistributors must be specified and this overrides the `reg-base` parameter (except that `reg-base` will still be used for the top-level redistributor). If `reg-base-per-redistributor-file` is specified, this parameter is ignored.

Type: `string`

Default value: ""

### **reg-base-per-redistributor-file**

Path to file containing base address for each redistributor in the form '0.0.0.0=0x2c010000, 0.0.0.1=0x2c020000'. All redistributors must be specified and this overrides the `reg-base` parameter (except that `reg-base` will still be used for the top-level redistributor). If this parameter is specified, `reg-base-per-redistributor` parameter will be ignored even when it is given.

Type: `string`

Default value: ""

**report-MSI-error-via-statusr**

Report MSI error via GITS\_STATUSR. (0:unsupported, 1:report by GITS\_STATUSR, 2:report by GITS\_STATUSR and interrupt as well).

Type: unsigned

Default value: 0

**rme\_default\_mecid\_nonsecure**

Default MECID value for NON-SECURE PAS.

Type: uint16\_t

Default value: 0

**sgi-range-selector-support**

Device has support for the Range Selector feature for SGI.

Type: bool

Default value: false

**single-set-support**

When true, forces redistributors to recall interrupts with a clear rather than issue a second Set command.

Type: bool

Default value: false

**supports-shareability**

Device supports shareability attributes on outgoing memory bus (i.e. is modelling an ACElite port rather than an AXI4 port).

Type: bool

Default value: true

**trace-speculative-lpi-property-update**

Trace LPI property updates performed on speculative accesses (useful for debugging LPI).

Type: bool

Default value: false

**vPE-table-entry-size-in-doubleword**

The size of one entry in double word of vPE configuration table. The value decremented by one is shown at GICR\_VPROPBASER.Entry\_Size. Current model mandates the minimum entry size to be 4 doublewords. When lower value is given, it is truncated to 4.

Type: `unsigned`

Default value: `5`

**virtual-lpi-support**

GICv4 Virtual LPIs and Direct injection of Virtual LPIs supported.

Type: `bool`

Default value: `false`

**virtual-priority-bits**

Number of implemented virtual priority bits.

Type: `uint8_t`

Default value: `5`

**wakeup-on-reset**

Go against specification and start redistributors in woken-up state at reset. This allows software that was written for previous versions of the GICv3/GICv4 specification to work correctly. This should not be used for production code or when the distributor is used separately from the core fast model.

Type: `bool`

Default value: `false`

## 3.200 GIC\_IRI\_Filter

Defined in `LISA/GIC_IRI_Filter.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

## About GIC\_IRI\_Filter

The GIC\_IRI\_Filter has similar behavior to the GIC\_IRI, except for the slave interface. Any transaction accessing a 4 KB page that is not used by the GIC, as configurable through the parameters, is forwarded to the `pvbus_filtermiss_m` port, which is only present in this variant.



Note

- Set the `FASTSIM_GIC_MEMORY_MAP` environment variable to 1 to print to stderr the memory map of any GICv3 or later models that are included in the platform being run.
- Set the `GICD_ITARGETSR-RAZWI` parameter to true for legacy, GICv2-style routing, where interrupts are routed to the first processor in the system.

## Iris and MTI instances for GIC\_IRI\_Filter

This model has the following Iris instances:

Name	Instance type
GIC_IRI_Filter	GIC_IRI
GIC_IRI_Filter.rd_0	GICv3RedistributorInternal
GIC_IRI_Filter.rd_0_0	GICv3RedistributorInternal
GIC_IRI_Filter.rd_0_0_0	GICv3RedistributorInternal
GIC_IRI_Filter.rd_0_0_0_0	GICv3Redistributor
GIC_IRI_Filter.rd_0_0_0_0.ExportTest.GIC_IRI_Filter.rd_0_0_0_0.pvbus_m[0].pvbusmaster	PVBusMaster
GIC_IRI_Filter.rd_t1	GICv3Distributor

This model has the following MTI trace components:

Name	Component type
GIC_IRI_Filter.rd_0	GICv3RedistributorInternal
GIC_IRI_Filter.rd_0_0	GICv3RedistributorInternal
GIC_IRI_Filter.rd_0_0_0	GICv3RedistributorInternal
GIC_IRI_Filter.rd_0_0_0_0	GICv3Redistributor
GIC_IRI_Filter.rd_0_0_0_0.ExportTest.GIC_IRI_Filter.rd_0_0_0_0.pvbus_m[0].pvbusmaster	PVBusMaster
GIC_IRI_Filter.rd_t1	GICv3Distributor

## Ports for GIC\_IRI\_Filter

Port	Direction	Protocol	Description
cfgsdisable	slave	Signal	Disable some SPIs signal.
extended_ppi_in_0	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 0.
extended_ppi_in_100	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 100.
extended_ppi_in_101	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 101.
extended_ppi_in_102	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 102.
extended_ppi_in_103	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 103.

Port	Direction	Protocol	Description
extended_ppi_in_104	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 104.
extended_ppi_in_105	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 105.
extended_ppi_in_106	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 106.
extended_ppi_in_107	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 107.
extended_ppi_in_108	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 108.
extended_ppi_in_109	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 109.
extended_ppi_in_10	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 10.
extended_ppi_in_110	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 110.
extended_ppi_in_111	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 111.
extended_ppi_in_112	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 112.
extended_ppi_in_113	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 113.
extended_ppi_in_114	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 114.
extended_ppi_in_115	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 115.
extended_ppi_in_116	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 116.
extended_ppi_in_117	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 117.
extended_ppi_in_118	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 118.
extended_ppi_in_119	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 119.
extended_ppi_in_11	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 11.
extended_ppi_in_120	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 120.
extended_ppi_in_121	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 121.
extended_ppi_in_122	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 122.
extended_ppi_in_123	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 123.
extended_ppi_in_124	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 124.
extended_ppi_in_125	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 125.
extended_ppi_in_126	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 126.
extended_ppi_in_127	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 127.
extended_ppi_in_128	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 128.
extended_ppi_in_129	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 129.
extended_ppi_in_12	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 12.
extended_ppi_in_130	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 130.
extended_ppi_in_131	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 131.
extended_ppi_in_132	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 132.
extended_ppi_in_133	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 133.
extended_ppi_in_134	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 134.
extended_ppi_in_135	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 135.
extended_ppi_in_136	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 136.
extended_ppi_in_137	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 137.
extended_ppi_in_138	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 138.
extended_ppi_in_139	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 139.
extended_ppi_in_13	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 13.

Port	Direction	Protocol	Description
extended_ppi_in_140	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 140.
extended_ppi_in_141	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 141.
extended_ppi_in_142	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 142.
extended_ppi_in_143	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 143.
extended_ppi_in_144	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 144.
extended_ppi_in_145	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 145.
extended_ppi_in_146	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 146.
extended_ppi_in_147	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 147.
extended_ppi_in_148	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 148.
extended_ppi_in_149	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 149.
extended_ppi_in_14	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 14.
extended_ppi_in_150	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 150.
extended_ppi_in_151	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 151.
extended_ppi_in_152	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 152.
extended_ppi_in_153	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 153.
extended_ppi_in_154	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 154.
extended_ppi_in_155	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 155.
extended_ppi_in_156	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 156.
extended_ppi_in_157	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 157.
extended_ppi_in_158	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 158.
extended_ppi_in_159	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 159.
extended_ppi_in_15	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 15.
extended_ppi_in_160	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 160.
extended_ppi_in_161	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 161.
extended_ppi_in_162	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 162.
extended_ppi_in_163	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 163.
extended_ppi_in_164	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 164.
extended_ppi_in_165	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 165.
extended_ppi_in_166	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 166.
extended_ppi_in_167	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 167.
extended_ppi_in_168	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 168.
extended_ppi_in_169	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 169.
extended_ppi_in_16	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 16.
extended_ppi_in_170	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 170.
extended_ppi_in_171	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 171.
extended_ppi_in_172	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 172.
extended_ppi_in_173	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 173.
extended_ppi_in_174	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 174.
extended_ppi_in_175	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 175.
extended_ppi_in_176	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 176.



Port	Direction	Protocol	Description
extended_ppi_in_177	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 177.
extended_ppi_in_178	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 178.
extended_ppi_in_179	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 179.
extended_ppi_in_17	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 17.
extended_ppi_in_180	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 180.
extended_ppi_in_181	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 181.
extended_ppi_in_182	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 182.
extended_ppi_in_183	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 183.
extended_ppi_in_184	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 184.
extended_ppi_in_185	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 185.
extended_ppi_in_186	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 186.
extended_ppi_in_187	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 187.
extended_ppi_in_188	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 188.
extended_ppi_in_189	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 189.
extended_ppi_in_18	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 18.
extended_ppi_in_190	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 190.
extended_ppi_in_191	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 191.
extended_ppi_in_192	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 192.
extended_ppi_in_193	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 193.
extended_ppi_in_194	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 194.
extended_ppi_in_195	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 195.
extended_ppi_in_196	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 196.
extended_ppi_in_197	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 197.
extended_ppi_in_198	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 198.
extended_ppi_in_199	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 199.
extended_ppi_in_19	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 19.
extended_ppi_in_1	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 1.
extended_ppi_in_200	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 200.
extended_ppi_in_201	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 201.
extended_ppi_in_202	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 202.
extended_ppi_in_203	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 203.
extended_ppi_in_204	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 204.
extended_ppi_in_205	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 205.
extended_ppi_in_206	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 206.
extended_ppi_in_207	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 207.
extended_ppi_in_208	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 208.
extended_ppi_in_209	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 209.
extended_ppi_in_20	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 20.
extended_ppi_in_210	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 210.
extended_ppi_in_211	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 211.

Port	Direction	Protocol	Description
extended_ppi_in_212	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 212.
extended_ppi_in_213	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 213.
extended_ppi_in_214	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 214.
extended_ppi_in_215	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 215.
extended_ppi_in_216	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 216.
extended_ppi_in_217	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 217.
extended_ppi_in_218	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 218.
extended_ppi_in_219	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 219.
extended_ppi_in_21	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 21.
extended_ppi_in_220	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 220.
extended_ppi_in_221	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 221.
extended_ppi_in_222	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 222.
extended_ppi_in_223	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 223.
extended_ppi_in_224	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 224.
extended_ppi_in_225	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 225.
extended_ppi_in_226	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 226.
extended_ppi_in_227	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 227.
extended_ppi_in_228	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 228.
extended_ppi_in_229	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 229.
extended_ppi_in_22	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 22.
extended_ppi_in_230	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 230.
extended_ppi_in_231	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 231.
extended_ppi_in_232	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 232.
extended_ppi_in_233	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 233.
extended_ppi_in_234	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 234.
extended_ppi_in_235	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 235.
extended_ppi_in_236	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 236.
extended_ppi_in_237	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 237.
extended_ppi_in_238	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 238.
extended_ppi_in_239	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 239.
extended_ppi_in_23	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 23.
extended_ppi_in_240	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 240.
extended_ppi_in_241	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 241.
extended_ppi_in_242	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 242.
extended_ppi_in_243	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 243.
extended_ppi_in_244	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 244.
extended_ppi_in_245	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 245.
extended_ppi_in_246	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 246.
extended_ppi_in_247	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 247.
extended_ppi_in_248	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 248.

Port	Direction	Protocol	Description
extended_ppi_in_249	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 249.
extended_ppi_in_24	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 24.
extended_ppi_in_250	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 250.
extended_ppi_in_251	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 251.
extended_ppi_in_252	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 252.
extended_ppi_in_253	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 253.
extended_ppi_in_254	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 254.
extended_ppi_in_255	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 255.
extended_ppi_in_25	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 25.
extended_ppi_in_26	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 26.
extended_ppi_in_27	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 27.
extended_ppi_in_28	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 28.
extended_ppi_in_29	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 29.
extended_ppi_in_2	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 2.
extended_ppi_in_30	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 30.
extended_ppi_in_31	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 31.
extended_ppi_in_32	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 32.
extended_ppi_in_33	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 33.
extended_ppi_in_34	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 34.
extended_ppi_in_35	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 35.
extended_ppi_in_36	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 36.
extended_ppi_in_37	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 37.
extended_ppi_in_38	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 38.
extended_ppi_in_39	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 39.
extended_ppi_in_3	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 3.
extended_ppi_in_40	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 40.
extended_ppi_in_41	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 41.
extended_ppi_in_42	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 42.
extended_ppi_in_43	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 43.
extended_ppi_in_44	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 44.
extended_ppi_in_45	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 45.
extended_ppi_in_46	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 46.
extended_ppi_in_47	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 47.
extended_ppi_in_48	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 48.
extended_ppi_in_49	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 49.
extended_ppi_in_4	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 4.
extended_ppi_in_50	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 50.
extended_ppi_in_51	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 51.
extended_ppi_in_52	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 52.
extended_ppi_in_53	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 53.

Port	Direction	Protocol	Description
extended_ppi_in_54	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 54.
extended_ppi_in_55	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 55.
extended_ppi_in_56	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 56.
extended_ppi_in_57	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 57.
extended_ppi_in_58	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 58.
extended_ppi_in_59	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 59.
extended_ppi_in_5	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 5.
extended_ppi_in_60	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 60.
extended_ppi_in_61	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 61.
extended_ppi_in_62	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 62.
extended_ppi_in_63	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 63.
extended_ppi_in_64	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 64.
extended_ppi_in_65	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 65.
extended_ppi_in_66	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 66.
extended_ppi_in_67	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 67.
extended_ppi_in_68	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 68.
extended_ppi_in_69	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 69.
extended_ppi_in_6	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 6.
extended_ppi_in_70	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 70.
extended_ppi_in_71	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 71.
extended_ppi_in_72	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 72.
extended_ppi_in_73	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 73.
extended_ppi_in_74	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 74.
extended_ppi_in_75	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 75.
extended_ppi_in_76	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 76.
extended_ppi_in_77	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 77.
extended_ppi_in_78	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 78.
extended_ppi_in_79	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 79.
extended_ppi_in_7	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 7.
extended_ppi_in_80	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 80.
extended_ppi_in_81	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 81.
extended_ppi_in_82	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 82.
extended_ppi_in_83	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 83.
extended_ppi_in_84	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 84.
extended_ppi_in_85	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 85.
extended_ppi_in_86	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 86.
extended_ppi_in_87	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 87.
extended_ppi_in_88	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 88.
extended_ppi_in_89	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 89.
extended_ppi_in_8	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 8.

Port	Direction	Protocol	Description
extended_ppi_in_90	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 90.
extended_ppi_in_91	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 91.
extended_ppi_in_92	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 92.
extended_ppi_in_93	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 93.
extended_ppi_in_94	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 94.
extended_ppi_in_95	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 95.
extended_ppi_in_96	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 96.
extended_ppi_in_97	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 97.
extended_ppi_in_98	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 98.
extended_ppi_in_99	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 99.
extended_ppi_in_9	slave	Signal	Extended private peripheral interrupts (ID1056-ID1119) for cpu 9.
extended_spi_in	slave	Signal	Extended Shared peripheral interrupts.
msi_error_interrupt	master	Signal	When report of MSI error allowed through interrupt, loop-back this signal to relevant IRQ input signal
po_reset	slave	Signal	Resets.
ppi_in_0	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 0.
ppi_in_100	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 100.
ppi_in_101	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 101.
ppi_in_102	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 102.
ppi_in_103	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 103.
ppi_in_104	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 104.
ppi_in_105	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 105.
ppi_in_106	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 106.
ppi_in_107	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 107.
ppi_in_108	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 108.
ppi_in_109	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 109.
ppi_in_10	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 10.
ppi_in_110	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 110.
ppi_in_111	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 111.
ppi_in_112	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 112.
ppi_in_113	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 113.
ppi_in_114	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 114.
ppi_in_115	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 115.
ppi_in_116	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 116.
ppi_in_117	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 117.
ppi_in_118	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 118.
ppi_in_119	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 119.
ppi_in_11	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 11.
ppi_in_120	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 120.
ppi_in_121	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 121.
ppi_in_122	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 122.

Port	Direction	Protocol	Description
ppi_in_123	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 123.
ppi_in_124	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 124.
ppi_in_125	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 125.
ppi_in_126	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 126.
ppi_in_127	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 127.
ppi_in_128	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 128.
ppi_in_129	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 129.
ppi_in_12	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 12.
ppi_in_130	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 130.
ppi_in_131	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 131.
ppi_in_132	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 132.
ppi_in_133	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 133.
ppi_in_134	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 134.
ppi_in_135	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 135.
ppi_in_136	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 136.
ppi_in_137	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 137.
ppi_in_138	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 138.
ppi_in_139	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 139.
ppi_in_13	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 13.
ppi_in_140	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 140.
ppi_in_141	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 141.
ppi_in_142	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 142.
ppi_in_143	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 143.
ppi_in_144	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 144.
ppi_in_145	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 145.
ppi_in_146	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 146.
ppi_in_147	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 147.
ppi_in_148	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 148.
ppi_in_149	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 149.
ppi_in_14	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 14.
ppi_in_150	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 150.
ppi_in_151	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 151.
ppi_in_152	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 152.
ppi_in_153	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 153.
ppi_in_154	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 154.
ppi_in_155	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 155.
ppi_in_156	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 156.
ppi_in_157	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 157.
ppi_in_158	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 158.
ppi_in_159	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 159.



Port	Direction	Protocol	Description
ppi_in_15	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 15.
ppi_in_160	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 160.
ppi_in_161	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 161.
ppi_in_162	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 162.
ppi_in_163	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 163.
ppi_in_164	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 164.
ppi_in_165	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 165.
ppi_in_166	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 166.
ppi_in_167	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 167.
ppi_in_168	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 168.
ppi_in_169	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 169.
ppi_in_16	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 16.
ppi_in_170	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 170.
ppi_in_171	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 171.
ppi_in_172	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 172.
ppi_in_173	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 173.
ppi_in_174	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 174.
ppi_in_175	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 175.
ppi_in_176	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 176.
ppi_in_177	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 177.
ppi_in_178	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 178.
ppi_in_179	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 179.
ppi_in_17	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 17.
ppi_in_180	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 180.
ppi_in_181	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 181.
ppi_in_182	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 182.
ppi_in_183	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 183.
ppi_in_184	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 184.
ppi_in_185	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 185.
ppi_in_186	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 186.
ppi_in_187	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 187.
ppi_in_188	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 188.
ppi_in_189	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 189.
ppi_in_18	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 18.
ppi_in_190	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 190.
ppi_in_191	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 191.
ppi_in_192	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 192.
ppi_in_193	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 193.
ppi_in_194	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 194.
ppi_in_195	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 195.

Port	Direction	Protocol	Description
ppi_in_196	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 196.
ppi_in_197	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 197.
ppi_in_198	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 198.
ppi_in_199	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 199.
ppi_in_19	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 19.
ppi_in_1	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 1.
ppi_in_200	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 200.
ppi_in_201	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 201.
ppi_in_202	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 202.
ppi_in_203	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 203.
ppi_in_204	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 204.
ppi_in_205	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 205.
ppi_in_206	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 206.
ppi_in_207	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 207.
ppi_in_208	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 208.
ppi_in_209	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 209.
ppi_in_20	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 20.
ppi_in_210	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 210.
ppi_in_211	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 211.
ppi_in_212	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 212.
ppi_in_213	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 213.
ppi_in_214	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 214.
ppi_in_215	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 215.
ppi_in_216	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 216.
ppi_in_217	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 217.
ppi_in_218	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 218.
ppi_in_219	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 219.
ppi_in_21	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 21.
ppi_in_220	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 220.
ppi_in_221	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 221.
ppi_in_222	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 222.
ppi_in_223	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 223.
ppi_in_224	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 224.
ppi_in_225	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 225.
ppi_in_226	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 226.
ppi_in_227	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 227.
ppi_in_228	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 228.
ppi_in_229	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 229.
ppi_in_22	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 22.
ppi_in_230	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 230.



Port	Direction	Protocol	Description
ppi_in_231	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 231.
ppi_in_232	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 232.
ppi_in_233	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 233.
ppi_in_234	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 234.
ppi_in_235	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 235.
ppi_in_236	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 236.
ppi_in_237	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 237.
ppi_in_238	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 238.
ppi_in_239	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 239.
ppi_in_23	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 23.
ppi_in_240	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 240.
ppi_in_241	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 241.
ppi_in_242	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 242.
ppi_in_243	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 243.
ppi_in_244	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 244.
ppi_in_245	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 245.
ppi_in_246	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 246.
ppi_in_247	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 247.
ppi_in_248	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 248.
ppi_in_249	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 249.
ppi_in_24	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 24.
ppi_in_250	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 250.
ppi_in_251	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 251.
ppi_in_252	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 252.
ppi_in_253	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 253.
ppi_in_254	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 254.
ppi_in_255	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 255.
ppi_in_25	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 25.
ppi_in_26	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 26.
ppi_in_27	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 27.
ppi_in_28	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 28.
ppi_in_29	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 29.
ppi_in_2	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 2.
ppi_in_30	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 30.
ppi_in_31	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 31.
ppi_in_32	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 32.
ppi_in_33	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 33.
ppi_in_34	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 34.
ppi_in_35	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 35.
ppi_in_36	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 36.

Port	Direction	Protocol	Description
ppi_in_37	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 37.
ppi_in_38	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 38.
ppi_in_39	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 39.
ppi_in_3	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 3.
ppi_in_40	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 40.
ppi_in_41	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 41.
ppi_in_42	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 42.
ppi_in_43	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 43.
ppi_in_44	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 44.
ppi_in_45	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 45.
ppi_in_46	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 46.
ppi_in_47	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 47.
ppi_in_48	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 48.
ppi_in_49	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 49.
ppi_in_4	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 4.
ppi_in_50	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 50.
ppi_in_51	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 51.
ppi_in_52	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 52.
ppi_in_53	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 53.
ppi_in_54	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 54.
ppi_in_55	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 55.
ppi_in_56	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 56.
ppi_in_57	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 57.
ppi_in_58	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 58.
ppi_in_59	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 59.
ppi_in_5	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 5.
ppi_in_60	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 60.
ppi_in_61	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 61.
ppi_in_62	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 62.
ppi_in_63	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 63.
ppi_in_64	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 64.
ppi_in_65	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 65.
ppi_in_66	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 66.
ppi_in_67	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 67.
ppi_in_68	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 68.
ppi_in_69	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 69.
ppi_in_6	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 6.
ppi_in_70	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 70.
ppi_in_71	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 71.
ppi_in_72	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 72.

Port	Direction	Protocol	Description
ppi_in_73	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 73.
ppi_in_74	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 74.
ppi_in_75	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 75.
ppi_in_76	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 76.
ppi_in_77	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 77.
ppi_in_78	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 78.
ppi_in_79	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 79.
ppi_in_7	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 7.
ppi_in_80	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 80.
ppi_in_81	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 81.
ppi_in_82	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 82.
ppi_in_83	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 83.
ppi_in_84	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 84.
ppi_in_85	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 85.
ppi_in_86	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 86.
ppi_in_87	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 87.
ppi_in_88	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 88.
ppi_in_89	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 89.
ppi_in_8	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 8.
ppi_in_90	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 90.
ppi_in_91	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 91.
ppi_in_92	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 92.
ppi_in_93	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 93.
ppi_in_94	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 94.
ppi_in_95	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 95.
ppi_in_96	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 96.
ppi_in_97	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 97.
ppi_in_98	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 98.
ppi_in_99	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 99.
ppi_in_9	slave	Signal	Private peripheral interrupts (ID16-ID31) for cpu 9.
pvbus_filtermiss_m	master	PVBus	Passthrough for accesses to pages not used by the GIC IRI.
pvbus_m	master	PVBus	Memory bus for transactions generated by the GIC.
pvbus_s	slave	PVBus	Memory bus in.
redistributor_m	master	GICv3Comms	Input from and output to CPU interface.
reset	slave	Signal	Resets.
spi_in	slave	Signal	Shared peripheral interrupts.
wake_request	master	Signal	Power management outputs.
wire_to_msi_in_0	slave	Signal	Wire-to-MSI interrupts for architectural consolidator 0.
wire_to_msi_in_1	slave	Signal	Wire-to-MSI interrupts for architectural consolidator 1.
wire_to_msi_in_2	slave	Signal	Wire-to-MSI interrupts for architectural consolidator 2.

Port	Direction	Protocol	Description
wire_to_msi_in_3	slave	Signal	Wire-to-MSI interrupts for architectural consolidator 3.

## Parameters for GIC\_IRI\_Filter

### A3-affinity-supported

Device supports affinity level 3 values that are non-zero.

Type: `bool`

Default value: `false`

### ARE-fixed-to-one

GICv2 compatibility is not supported and GICD\_CTLR.ARE\_\* is always one.

Type: `bool`

Default value: `false`

### CPU-affinities

A comma separated list of dotted quads containing the affinities of all PEs connected to this IRI. If CPU-affinities-file is specified, this parameter is ignored.

Type: `string`

Default value: `N/A`

### CPU-affinities-file

A file containing comma separated list of dotted quads containing the affinities of all PEs connected to this IRI. If this parameter is specified, CPU-affinities parameter will be ignored even when it is given.

Type: `string`

Default value: `N/A`

### DPG-ARE-only

Limit application of DPG bits to interrupt groups for which ARE=1.

Type: `bool`

Default value: `false`

### DPG-bits-implemented

Enable implementation of interrupt group participation bits or DPG bits in GICR\_CTLR.

Type: `bool`

Default value: `false`

### **DS-fixed-to-zero**

Enable/disable support of single security state.

Type: `bool`

Default value: `false`

### **GICD-alias**

In GICv2 mode: the base address for a 4k page alias of the first 4k of the Distributor page, in GICv3/GICv4 mode. the base address of a 64KB page containing message based SPI signalling register aliases(0:Disabled).

Type: `uint64_t`

Default value: `0x0`

### **GICD-legacy-registers-as-reserved**

When ARE is **RAO/WI**, makes superfluous registers in GICD reserved (including for the purpose of STATUSR updates).

Type: `bool`

Default value: `false`

### **GICD\_CTLR-DS-1-means-secure-only**

If GICD\_CTLR.DS=1, GICD supports a single security state which is secure if this is true, otherwise is non-secure.

Type: `bool`

Default value: `false`

### **GICD\_ITARGETSR-RAZWI**

If true, the GICD\_ITARGETS registers are **RAZ/WI**.

Type: `bool`

Default value: `false`

### **GICD\_PIDR**

The value for the GICD\_PIDR registers, if non-zero. Note: fixed fields (device type etc.) will be overridden in this value.

Type: `uint64_t`

Default value: `0x0`

**GICD\_TYPER2**

GICD\_TYPER2 value containing VID and VIL to define the width of vPEID for GICv4.1.

Type: `uint32_t`

Default value: 0

**GICR-clear-enable-supported**

When true, this sets the value of the RO bit GICR\_CTLR.CES with the value of the parameter `allow-LPIEN-clear`, making it visible to software.

Type: `bool`

Default value: `false`

**GICR-invalidate-registers-implemented**

When true, the registers GICR\_INVLPIR, GICR\_INVALLR and GICR\_SYNCRR are implemented.

Type: `bool`

Default value: `false`

**GICR\_PIDR**

The value for the GICR\_PIDR registers, if non-zero. Note: fixed fields (device type etc.) will be overridden in this value.

Type: `uint64_t`

Default value: 0x0

**GICR\_PROPBASER-read-only**

GICR\_PROPBASER register is read-only.

Type: `bool`

Default value: `false`

**GICR\_PROPBASER-reset-value**

Value of GICR\_PROPBASER on reset.

Type: `uint64_t`

Default value: 0x0

**GITS\_BASER0-entry-bytes**

Number of bytes required per entry for GITS\_BASER0 register.

Type: `unsigned`

Default value: 8

#### **GITS\_BASER0-indirect-raz**

Indirect field for GITS\_BASER0 register is **RAZ/WI**.

Type: `bool`

Default value: `false`

#### **GITS\_BASER0-type**

Type field for GITS\_BASER0 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: `uint8_t`

Default value: 0

#### **GITS\_BASER1-entry-bytes**

Number of bytes required per entry for GITS\_BASER1 register.

Type: `unsigned`

Default value: 8

#### **GITS\_BASER1-indirect-raz**

Indirect field for GITS\_BASER1 register is **RAZ/WI**.

Type: `bool`

Default value: `false`

#### **GITS\_BASER1-type**

Type field for GITS\_BASER1 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: `uint8_t`

Default value: 0

#### **GITS\_BASER2-entry-bytes**

Number of bytes required per entry for GITS\_BASER2 register.

Type: `unsigned`

Default value: 8

**GITS\_BASER2-indirect-RAZ**

Indirect field for GITS\_BASER2 register is **RAZ/WI**.

Type: `bool`

Default value: `false`

**GITS\_BASER2-type**

Type field for GITS\_BASER2 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: `uint8_t`

Default value: 0

**GITS\_BASER3-entry-bytes**

Number of bytes required per entry for GITS\_BASER3 register.

Type: `unsigned`

Default value: 8

**GITS\_BASER3-indirect-RAZ**

Indirect field for GITS\_BASER3 register is **RAZ/WI**.

Type: `bool`

Default value: `false`

**GITS\_BASER3-type**

Type field for GITS\_BASER3 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: `uint8_t`

Default value: 0

**GITS\_BASER4-entry-bytes**

Number of bytes required per entry for GITS\_BASER4 register.

Type: `unsigned`

Default value: 8

**GITS\_BASER4-indirect-RAZ**

Indirect field for GITS\_BASER4 register is **RAZ/WI**.



Type: `bool`

Default value: `false`

#### **GITS\_BASER4-type**

Type field for GITS\_BASER4 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: `uint8_t`

Default value: 0

#### **GITS\_BASER5-entry-bytes**

Number of bytes required per entry for GITS\_BASER5 register.

Type: `unsigned`

Default value: 8

#### **GITS\_BASER5-indirect-RAZ**

Indirect field for GITS\_BASER5 register is **RAZ/WI**.

Type: `bool`

Default value: `false`

#### **GITS\_BASER5-type**

Type field for GITS\_BASER5 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: `uint8_t`

Default value: 0

#### **GITS\_BASER6-entry-bytes**

Number of bytes required per entry for GITS\_BASER6 register.

Type: `unsigned`

Default value: 8

#### **GITS\_BASER6-indirect-RAZ**

Indirect field for GITS\_BASER6 register is **RAZ/WI**.

Type: `bool`

Default value: `false`

**GITS\_BASER6-type**

Type field for GITS\_BASER6 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: `uint8_t`

Default value: 0

**GITS\_BASER7-entry-bytes**

Number of bytes required per entry for GITS\_BASER7 register.

Type: `unsigned`

Default value: 8

**GITS\_BASER7-indirect-RAZ**

Indirect field for GITS\_BASER7 register is **RAZ/WI**.

Type: `bool`

Default value: `false`

**GITS\_BASER7-type**

Type field for GITS\_BASER7 register. 0 = Unimplemented; 1 = Devices; 2 = Virtual Processors; 3 = Physical Processors; 4 = Collections.

Type: `uint8_t`

Default value: 0

**GITS\_PIDR**

The value for the GITS\_PIDR registers, if non-zero. Note: fixed fields (device type etc.) will be overridden in this value.

Type: `uint64_t`

Default value: 0x0

**ICFGR-PPI-mask**

Mask for writes to ICFGR registers that configure PPIs.

Type: `uint32_t`

Default value: 0xaaaaaaaa

**ICFGR-PPI-reset**

Reset value for ICFGR registers that configure PPIs.

Type: `uint32_t`

Default value: `0x0`

#### **ICFGR-SGI-mask**

Mask for writes to ICFGR registers that configure SGIs.

Type: `uint32_t`

Default value: `0x0`

#### **ICFGR-SGI-reset**

Reset value for ICFGR registers that configure SGIs.

Type: `uint32_t`

Default value: `0xaaaaaaaa`

#### **ICFGR-SPI-mask**

Mask for writes to ICFGR registers that configure SPIs.

Type: `uint32_t`

Default value: `0xaaaaaaaa`

#### **ICFGR-SPI-reset**

Reset value for ICFGR registers that configure SPIs.

Type: `uint32_t`

Default value: `0x0`

#### **ICFGR-rsvd-bit**

If ARE=0, the value of reserved bits i.e. bit 0,2,4..30 of ICFGR<sub>n</sub> for n>0.

Type: `bool`

Default value: `false`

#### **IGROUP-PPI-mask**

Mask for writes to PPI bits in IGROUP registers.

Type: `uint16_t`

Default value: `0xffff`

#### **IGROUP-PPI-reset**

Reset value for SGI bits in IGROUP registers.

Type: `uint16_t`

Default value: `0x0`

### **IGROUP-SGI-mask**

Mask for writes to SGI bits in IGROUP registers.

Type: `uint16_t`

Default value: `0xffff`

### **IGROUP-SGI-reset**

Reset value for SGI bits in IGROUP registers.

Type: `uint16_t`

Default value: `0x0`

### **IIDR**

GICD\_IIDR and GICR\_IIDR value.

Type: `uint32_t`

Default value: `0x0`

### **IRI-ID-bits**

Number of bits used to represent interrupts IDs in the Distributor and Redistributors, forced to 10 when none of LPIs, extended SPLs or extended PPIs is supported.

Type: `int`

Default value: `16`

### **IROUTER-IRM-RAZ-WI**

GICD\_IROUTERn.InterruptRoutingMode is **RAZ/WI**.

Type: `bool`

Default value: `false`

### **ITS-BASER-force-page-alignement**

Force alignment of address written to a GITS\_BASER register to the page size configured.

Type: `bool`

Default value: `true`

**ITS-ID-bits**

Number of interrupt bits supported by ITS.

Type: `uint8_t`

Default value: 16

**ITS-MOVALL-update-collections**

Whether MOVALL command updates the collection entires.

Type: `bool`

Default value: `false`

**ITS-TRANSLATE64R**

Add an implementation specific register at 0x10008 supporting 64 bit TRANSLATER (dev[63:32], interrupt[31:0]).

Type: `bool`

Default value: `false`

**ITS-cache-invalidate-on-disable**

Sets the RO bit GITS\_TYPER.INV. When true, after the following sequence: 1) GITS\_CTLR.Enabled written 1->0, 2) GITS\_CTLR.Quiescent observed as 1, 3) GITS\_BASER<n>.Valid written 1->0, there is no cached information from the ITS memory structure pointed to by GITS\_BASER<n>.

Type: `bool`

Default value: `false`

**ITS-collection-ID-bits**

Number of collection bits supported by ITS (optional parameter, 0 => 16bits support and GITS\_TYPER.CIL=0).

Type: `int`

Default value: 0

**ITS-count**

Number of Interrupt Translation Services to be instantiated (0=none).

Type: `uint8_t`

Default value: 0

**ITS-cumulative-collection-tables**

When true, the supported amount of collections is the sum of GITS\_TYPER.HCC and the number of collections supported in memory, otherwise, simply the number supported in memory only. Irrelevant when HCC=0.

Type: `bool`

Default value: `true`

**ITS-device-bits**

Number of bits supported for ITS device IDs.

Type: `int`

Default value: `16`

**ITS-enable-itt-address-verification**

If true, a transaction will be sent to ITT Address for verification.

Type: `bool`

Default value: `false`

**ITS-entry-size**

Number of bytes required to store each entry in the ITT tables.

Type: `int`

Default value: `8`

**ITS-hardware-collection-count**

Number of hardware collections held exclusively in the ITS.

Type: `int`

Default value: `0`

**ITS-legacy-iidr-typer-offset**

Put the GITS\_IIDR and GITS\_TYPER registers at their older offset of 0x8 and 0x4 respectively.

Type: `bool`

Default value: `false`

**ITS-shared-vPE-table**

Number of affinity levels to which the vPE configuration table is shared. This parameter is valid when `has-gicv4.1` is true.

Type: `uint8_t`

Default value: 0

### **ITS-threaded-command-queue**

Enable execution of ITS commands in a separate thread which is sometimes required for cosimulation.

Type: `bool`

Default value: `true`

### **ITS-use-physical-target-addresses**

Use physical hardware addresses for targets in ITS commands – must be true for distributed implementations.

Type: `bool`

Default value: `true`

### **ITS-vmovp-bit**

Device supports software issuing a VMOVP to only one of the ITSs that has a mapping for a vPE. The device itself ensures synchronization of the VMOVP command across all ITSs that have mapping for that vPE.

Type: `bool`

Default value: `false`

### **ITS0-base**

Register base address for ITS0 (automatic if 0).

Type: `uint64_t`

Default value: 0

### **ITS1-base**

Register base address for ITS1 (automatic if 0).

Type: `uint64_t`

Default value: 0

### **ITS2-base**

Register base address for ITS2 (automatic if 0).

Type: `uint64_t`

Default value: 0

**ITS3-base**

Register base address for ITS3 (automatic if 0).

Type: `uint64_t`

Default value: 0

**LPI-cache-check-data**

Enable Cached LPI data against memory checking when available for cache type.

Type: `bool`

Default value: `false`

**LPI-cache-type**

Cache type for LPIs, 0:No caching, 1:Full caching.

Type: `uint8_t`

Default value: 1

**MSI\_IIDR**

Value returned in MSI\_IIDR registers.

Type: `uint32_t`

Default value: 0x0

**MSI\_NS-frame0-base**

If non-zero, sets the base address used for non-secure MSI frame 0 registers.

Type: `uint64_t`

Default value: 0x0

**MSI\_NS-frame0-max-SPI**

Maximum SPI ID supported by non-secure MSI frame 0. Set to 0 to disable frame.

Type: `uint16_t`

Default value: 0

**MSI\_NS-frame0-min-SPI**

Minimum SPI ID supported by non-secure MSI frame 0. Set to 0 to disable frame.

Type: `uint16_t`



Default value: 0

**MSI\_NS-frame1-base**

If non-zero, sets the base address used for non-secure MSI frame 1 registers.

Type: uint64\_t

Default value: 0x0

**MSI\_NS-frame1-max-SPI**

Maximum SPI ID supported by non-secure MSI frame 1. Set to 0 to disable frame.

Type: uint16\_t

Default value: 0

**MSI\_NS-frame1-min-SPI**

Minimum SPI ID supported by non-secure MSI frame 1. Set to 0 to disable frame.

Type: uint16\_t

Default value: 0

**MSI\_NS-frame2-base**

If non-zero, sets the base address used for non-secure MSI frame 2 registers.

Type: uint64\_t

Default value: 0x0

**MSI\_NS-frame2-max-SPI**

Maximum SPI ID supported by non-secure MSI frame 2. Set to 0 to disable frame.

Type: uint16\_t

Default value: 0

**MSI\_NS-frame2-min-SPI**

Minimum SPI ID supported by non-secure MSI frame 2. Set to 0 to disable frame.

Type: uint16\_t

Default value: 0

**MSI\_NS-frame3-base**

If non-zero, sets the base address used for non-secure MSI frame 3 registers.

Type: uint64\_t

Default value: 0x0

**MSI\_NS-frame3-max-SPI**

Maximum SPI ID supported by non-secure MSI frame 3. Set to 0 to disable frame.

Type: uint16\_t

Default value: 0

**MSI\_NS-frame3-min-SPI**

Minimum SPI ID supported by non-secure MSI frame 3. Set to 0 to disable frame.

Type: uint16\_t

Default value: 0

**MSI\_NS-frame4-base**

If non-zero, sets the base address used for non-secure MSI frame 4 registers.

Type: uint64\_t

Default value: 0x0

**MSI\_NS-frame4-max-SPI**

Maximum SPI ID supported by non-secure MSI frame 4. Set to 0 to disable frame.

Type: uint16\_t

Default value: 0

**MSI\_NS-frame4-min-SPI**

Minimum SPI ID supported by non-secure MSI frame 4. Set to 0 to disable frame.

Type: uint16\_t

Default value: 0

**MSI\_NS-frame5-base**

If non-zero, sets the base address used for non-secure MSI frame 5 registers.

Type: uint64\_t

Default value: 0x0

**MSI\_NS-frame5-max-SPI**

Maximum SPI ID supported by non-secure MSI frame 5. Set to 0 to disable frame.

Type: uint16\_t

Default value: 0

**MSI\_NS-frame5-min-SPI**

Minimum SPI ID supported by non-secure MSI frame 5. Set to 0 to disable frame.

Type: uint16\_t

Default value: 0

**MSI\_NS-frame6-base**

If non-zero, sets the base address used for non-secure MSI frame 6 registers.

Type: uint64\_t

Default value: 0x0

**MSI\_NS-frame6-max-SPI**

Maximum SPI ID supported by non-secure MSI frame 6. Set to 0 to disable frame.

Type: uint16\_t

Default value: 0

**MSI\_NS-frame6-min-SPI**

Minimum SPI ID supported by non-secure MSI frame 6. Set to 0 to disable frame.

Type: uint16\_t

Default value: 0

**MSI\_NS-frame7-base**

If non-zero, sets the base address used for non-secure MSI frame 7 registers.

Type: uint64\_t

Default value: 0x0

**MSI\_NS-frame7-max-SPI**

Maximum SPI ID supported by non-secure MSI frame 7. Set to 0 to disable frame.

Type: uint16\_t

Default value: 0

**MSI\_NS-frame7-min-SPI**

Minimum SPI ID supported by non-secure MSI frame 7. Set to 0 to disable frame.

Type: uint16\_t

Default value: 0

### **MSI\_PIDR**

The value for the MSI\_PIDR registers, if non-zero and distributor supports GICv2m. Note: fixed fields (device type etc.) will be overridden in this value.

Type: uint64\_t

Default value: 0x0

### **MSI\_S-frame0-base**

If non-zero, sets the base address used for secure MSI frame 0 registers.

Type: uint64\_t

Default value: 0x0

### **MSI\_S-frame0-max-SPI**

Maximum SPI ID supported by secure MSI frame 0. Set to 0 to disable frame.

Type: uint16\_t

Default value: 0

### **MSI\_S-frame0-min-SPI**

Minimum SPI ID supported by secure MSI frame 0. Set to 0 to disable frame.

Type: uint16\_t

Default value: 0

### **MSI\_S-frame1-base**

If non-zero, sets the base address used for secure MSI frame 1 registers.

Type: uint64\_t

Default value: 0x0

### **MSI\_S-frame1-max-SPI**

Maximum SPI ID supported by secure MSI frame 1. Set to 0 to disable frame.

Type: uint16\_t

Default value: 0

### **MSI\_S-frame1-min-SPI**

Minimum SPI ID supported by secure MSI frame 1. Set to 0 to disable frame.

Type: `uint16_t`

Default value: 0

### **MSI\_S-frame2-base**

If non-zero, sets the base address used for secure MSI frame 2 registers.

Type: `uint64_t`

Default value: 0x0

### **MSI\_S-frame2-max-SPI**

Maximum SPI ID supported by secure MSI frame 2. Set to 0 to disable frame.

Type: `uint16_t`

Default value: 0

### **MSI\_S-frame2-min-SPI**

Minimum SPI ID supported by secure MSI frame 2. Set to 0 to disable frame.

Type: `uint16_t`

Default value: 0

### **MSI\_S-frame3-base**

If non-zero, sets the base address used for secure MSI frame 3 registers.

Type: `uint64_t`

Default value: 0x0

### **MSI\_S-frame3-max-SPI**

Maximum SPI ID supported by secure MSI frame 3. Set to 0 to disable frame.

Type: `uint16_t`

Default value: 0

### **MSI\_S-frame3-min-SPI**

Minimum SPI ID supported by secure MSI frame 3. Set to 0 to disable frame.

Type: `uint16_t`

Default value: 0

### **MSI\_S-frame4-base**

If non-zero, sets the base address used for secure MSI frame 4 registers.

Type: uint64\_t

Default value: 0x0

#### **MSI\_S-frame4-max-SPI**

Maximum SPI ID supported by secure MSI frame 4. Set to 0 to disable frame.

Type: uint16\_t

Default value: 0

#### **MSI\_S-frame4-min-SPI**

Minimum SPI ID supported by secure MSI frame 4. Set to 0 to disable frame.

Type: uint16\_t

Default value: 0

#### **MSI\_S-frame5-base**

If non-zero, sets the base address used for secure MSI frame 5 registers.

Type: uint64\_t

Default value: 0x0

#### **MSI\_S-frame5-max-SPI**

Maximum SPI ID supported by secure MSI frame 5. Set to 0 to disable frame.

Type: uint16\_t

Default value: 0

#### **MSI\_S-frame5-min-SPI**

Minimum SPI ID supported by secure MSI frame 5. Set to 0 to disable frame.

Type: uint16\_t

Default value: 0

#### **MSI\_S-frame6-base**

If non-zero, sets the base address used for secure MSI frame 6 registers.

Type: uint64\_t

Default value: 0x0

#### **MSI\_S-frame6-max-SPI**

Maximum SPI ID supported by secure MSI frame 6. Set to 0 to disable frame.

Type: `uint16_t`

Default value: 0

#### **MSI\_S-frame6-min-SPI**

Minimum SPI ID supported by secure MSI frame 6. Set to 0 to disable frame.

Type: `uint16_t`

Default value: 0

#### **MSI\_S-frame7-base**

If non-zero, sets the base address used for secure MSI frame 7 registers.

Type: `uint64_t`

Default value: 0x0

#### **MSI\_S-frame7-max-SPI**

Maximum SPI ID supported by secure MSI frame 7. Set to 0 to disable frame.

Type: `uint16_t`

Default value: 0

#### **MSI\_S-frame7-min-SPI**

Minimum SPI ID supported by secure MSI frame 7. Set to 0 to disable frame.

Type: `uint16_t`

Default value: 0

#### **PA\_SIZE**

Number of valid bits in physical address.

Type: `int`

Default value: 48

#### **PPI-implemented-mask**

Mask of PPIs that are implemented. One bit per PPI bit 0 == PPI 16 (first PPI). This will affect other masks.

Type: `uint16_t`

Default value: 0xffff

**SPI-count**

Number of SPIs that are implemented.

Type: `uint16_t`

Default value: `224`

**SPI-message-based-support**

Distributor supports message based signaling of SPI.

Type: `bool`

Default value: `true`

**SPI-unimplemented**

A comma separated list of unimplemented SPIs ranges for sparse SPI definition(for ex: '35, 39-42, 73').

Type: `string`

Default value: `""`

**STATUSR-implemented**

Determines whether the GICR\_STATUSR register is implemented.

Type: `bool`

Default value: `true`

**add-output-cpu-wake-request-signal-from-redistributor**

If true, the redistributor will have the output signal `cpu_wake_request` from GIC to DSU and if false, the signals are not added to the redistributor.

Type: `bool`

Default value: `false`

**allow-LPIEN-clear**

Allow RW behaviour on GICR\_CTLR.LPIEN instead of set once.

Type: `bool`

Default value: `false`

**clear-ISPENDR-bit-for-level-sensitive-interrupt-when-acknowledged**

If true, acknowledgement of a level sensitive interrupt clears the corresponding bit in the ISPENDR register.



Type: `bool`

Default value: `false`

### **common-lpi-configuration**

Describes which re-distributors share (and must be configured with the same) LPI configuration table as described in GICR\_TYPER( 0:All, 1:A.x.x.x, 2:A.B.x.x, 3:A.B.C.x).

Type: `int`

Default value: `0`

### **common-vPE-table-affinity**

Affinity value list in the form of 'a.b.c.d, e.f.g.h, etc' given to the ITS(s) (where a.b.c.d corresponds to ITS0, e.f.g.h corresponds to ITS1 and so on). Under an affinity value the vPE configuration table is shared among redistributors where the level to be shared is defined by ITS-shared-vPE-table. This parameter is valid when has-gicv4.1 is true.

Type: `string`

Default value: `""`

### **consolidators**

Specify consolidators' base addresses, interrupt line counts and base interrupt IDs, in the form 'baseAddr0:itlineCount0:baseINTID0, baseAddr1:itlineCount1:baseINTID1, [etc]' (eg '0x3f100000:64:4096, 0x3f200000:64:4224'). The consolidators' count is inferred from the list (maximum of 4). If not specified, the component contains no consolidators.

Type: `string`

Default value: `""`

### **delay-ITS-accesses**

Delay accesses from the ITS until GICR\_SYNCRR is read.

Type: `bool`

Default value: `true`

### **delay-redistributor-accesses**

Delay memory accesses from the redistributor until GICR\_SYNCRR is read.

Type: `bool`

Default value: `true`

### **direct-lpi-support**

Enable support for LPI operations through GICR registers.

Type: `bool`

Default value: `false`

### **`enable_protocol_checking`**

Enable/disable protocol checking at cpu interface.

Type: `bool`

Default value: `false`

### **`enabled`**

Enable GICv3 functionality; when false the component is inactive.

Type: `bool`

Default value: `true`

### **`extended-ppi-count`**

Number of extended PPI supported.

Type: `unsigned`

Default value: 0

### **`extended-spi-count`**

Number of extended SPI supported.

Type: `unsigned`

Default value: 0

### **`fixed-routed-spis`**

Value of IROUTER[n] register in the form 'n=a.b.c.d, n='. *The RM bit of IROUTER is 0 when n=a.b.c.d is used else 1 when n= is used.* n can be  $\geq 32$  and  $\leq 1019$ .

Type: `string`

Default value: ""

### **`gicr-icfgr-extended-count`**

Number of extended GICR\_ICFGR registers supported.

Type: `uint8_t`

Default value: 4

**gicv2-only**

If true, when using the GICv3/GICv4 model, pretend to be a GICv2 system.

Type: `bool`

Default value: `false`

**group-enables-control-doorbell**

When true, GICR\_VPENDBASER.{VGrp0En,VGrp1En} are cached to allow GIC to check group enables when virtual interrupt targeting this VCPU which is non-resident reaches Redistributor.

Type: `bool`

Default value: `false`

**has-gicv4.1**

Enable GICv4.1 functionality; when false the component is inactive.

Type: `bool`

Default value: `false`

**has-two-security-states**

If true, has two security states.

Type: `bool`

Default value: `true`

**has\_VPENDBASER-dirty-flag-on-load**

GICR\_VPENDBASER.Dirty reflects transient loading state when valid=1.

Type: `bool`

Default value: `false`

**has\_mpam**

Implement ARMv8.4 MPAM Registers and associated functionality.values of this parameter are:- 0, feature is not enabled.- 1, feature is implemented if ARMv8.4 is enabled.- 2, feature is implemented.

Type: `uint8_t`

Default value: `0`

**has\_nmi**

Enable support for Non-maskable Interrupts (NMIs).

Type: `bool`

Default value: `false`

### **ignore-generate-sgi-when-no-are**

Ignore GenerateSGL packets coming from the CPU interface if both ARE\_S and ARE\_NS are 0.

Type: `bool`

Default value: `false`

### **individual-doorbell-not-supported**

For IRI with support of virtual interrupt, individual doorbell is not supported when true.

Type: `bool`

Default value: `false`

### **irouter-default-mask**

Default Mask value for IROUTER[32..1019] register in the form 'a.b.c.d'.

Type: `string`

Default value: `""`

### **irouter-default-reset**

Default Reset Value of IROUTER[32..1019] register in the form 'a.b.c.d' or '\*'.

Type: `string`

Default value: `N/A`

### **irouter-mask-values**

Mask Value of IROUTER[n] register in the form 'n=a.b.c.d'. n can be  $\geq 32$  and  $\leq 1019$ .

Type: `string`

Default value: `""`

### **irouter-reset-values**

Reset Value of IROUTER[n] register in the form 'n=a.b.c.d or n=\*'. n can be  $\geq 32$  and  $\leq 1019$ .

Type: `string`

Default value: `N/A`

### **legacy-sgi-enable-rao**

Enables for SGL associated with an ARE=0 regime are **RAO/WI**.

Type: `bool`

Default value: `false`

### **local-SEIs**

Generate SEI to signal internal issues.

Type: `bool`

Default value: `false`

### **local-VSEIs**

Generate VSEI to signal internal issues.

Type: `bool`

Default value: `false`

### **lockable-SPI-count**

Number of SPIs that are locked down when CFGSDISABLE signal is asserted. Only applies for GICv2.

Type: `uint8_t`

Default value: `0`

### **monolithic**

Indicate that the implementation is not distributed.

Type: `bool`

Default value: `false`

### **mpam\_max\_partid**

MPAM Maximum PARTID Supported.

Type: `uint16_t`

Default value: `0xffff`

### **mpam\_max\_pmg**

MPAM Maximum PMG Supported.

Type: `uint8_t`

Default value: `255`

**non-ARE-core-count**

Maximum number of non-ARE cores; normally used to pass the cluster-level NUM\_CORES parameter to the top-level redistributor.

Type: `int`

Default value: `8`

**outer-cacheability-support**

Allow configuration of outer cachability attributes in ITS and Redistributor.

Type: `bool`

Default value: `false`

**output\_attributes**

User-defined transform to be applied to bus attributes like ManagerID, ExtendedID or UserFlags. Currently, only works for MPAM Attributes encoding into bus attributes.

Type: `string`

Default value: `"ExtendedID[62:55]=MPAM_PMG, ExtendedID[54:39]=MPAM_PARTID, ExtendedID[38]=MPAM_NS"`

**print-memory-map**

Print memory map to stdout.

Type: `bool`

Default value: `false`

**priority-bits**

Number of implemented priority bits.

Type: `uint8_t`

Default value: `5`

**processor-numbers**

Specify processor numbers (as appears in GICR\_TYPER) in the form 0.0.0.0=0,0.0.0.1=1 etc.) If not specified, will number processors starting at 0.

Type: `string`

Default value: `""`

**redistributor-threaded-sync**

Enable execution of redistributor delayed transactions in a separate thread which is sometimes required for cosimulation.

Type: `bool`

Default value: `true`

**reg-base**

Base for decoding GICv3/GICv4 registers.

Type: `uint64_t`

Default value: `0x2c010000`

**reg-base-per-redistributor**

Base address for each redistributor in the form:

```
0.0.0.0=0x2c010000, 0.0.0.1=0x2c020000
```

All redistributors must be specified and this overrides the `reg-base` parameter (except that `reg-base` will still be used for the top-level redistributor). If `reg-base-per-redistributor-file` is specified, this parameter is ignored.

Type: `string`

Default value: `""`

**reg-base-per-redistributor-file**

Path to file containing the base address for each redistributor in the form '`0.0.0.0=0x2c010000, 0.0.0.1=0x2c020000`'. All redistributors must be specified and this overrides the `reg-base` parameter (except that `reg-base` will still be used for the top-level redistributor). If this parameter is specified, `reg-base-per-redistributor` parameter will be ignored even when it is given.

Type: `string`

Default value: `""`

**report-MSI-error-via-statusr**

Report MSI error via GITS\_STATUSR. (0:unsupported, 1:report by GITS\_STATUSR, 2:report by GITS\_STATUSR and interrupt as well).

Type: `unsigned`

Default value: `0`

**rme\_default\_mecid\_nonsecure**

Default MECID value for NON-SECURE PAS.

Type: `uint16_t`

Default value: `0`

**sgi-range-selector-support**

Device has support for the Range Selector feature for SGI.

Type: `bool`

Default value: `false`

**single-set-support**

When true, forces redistributors to recall interrupts with a clear rather than issue a second Set command.

Type: `bool`

Default value: `false`

**supports-shareability**

Device supports shareability attributes on outgoing memory bus (i.e. is modelling an ACElite port rather than an AXI4 port).

Type: `bool`

Default value: `true`

**trace-speculative-lpi-property-update**

Trace LPI property updates performed on speculative accesses (useful for debugging LPI).

Type: `bool`

Default value: `false`

**vPE-table-entry-size-in-doubleword**

The size of one entry in double word of vPE configuration table. The value decremented by one is shown at `GICR_VPROPBASER.Entry_Size`. Current model mandates the minimum entry size to be 4 doublewords. When lower value is given, it is truncated to 4.

Type: `unsigned`

Default value: `5`



**virtual-lpi-support**

GICv4 Virtual LPIs and Direct injection of Virtual LPIs supported.

Type: `bool`

Default value: `false`

**virtual-priority-bits**

Number of implemented virtual priority bits.

Type: `uint8_t`

Default value: `5`

**wakeup-on-reset**

Go against specification and start redistributors in woken-up state at reset. This allows software that was written for previous versions of the GICv3/GICv4 specification to work correctly. This should not be used for production code or when the distributor is used separately from the core fast model.

Type: `bool`

Default value: `false`

## 3.201 GICv3CommsLogger

Defined in `LISA/GICv3CommsLogger.lisa`.

**About GICv3CommsLogger**

Traces GICv3Comms activity.

**Iris and MTI instances for GICv3CommsLogger**

This model has the following Iris instances:

Name	Instance type
GICv3CommsLogger	GICv3CommsLogger

This model has the following MTI trace components:

Name	Component type
GICv3CommsLogger	GICv3CommsLogger

**Ports for GICv3CommsLogger**

Port	Direction	Protocol	Description
to_cpu	master	GICv3Comms	To connect to CPU.

Port	Direction	Protocol	Description
to_gic	slave	GICv3Comms	To connect to GIC.

### Parameters for GICv3CommsLogger

#### **verbose**

Print tracing information to attached debugger in addition to via MTI.

Type: `bool`

Default value: `false`

## 3.202 GICv3CommsPVBUS

Defined in `LISA/GICv3CommsPVBUS.lisa`.

### About GICv3CommsPVBUS

GICv3 Component for conversion between GICv3Comms protocol and PVBUS.

### Iris and MTI instances for GICv3CommsPVBUS

This model has the following Iris instances:

Name	Instance type
GICv3CommsPVBUS	GICv3CommsPVBUS
GICv3CommsPVBUS.ExportTest.GICv3CommsPVBUS.pvbus_m[0].pvbusmaster	PVBUSMaster
GICv3CommsPVBUS.bus_slave	PVBUSSlave

This model has the following MTI trace components:

Name	Component type
GICv3CommsPVBUS	GICv3CommsPVBUS
GICv3CommsPVBUS.ExportTest.GICv3CommsPVBUS.pvbus_m[0].pvbusmaster	PVBUSMaster
GICv3CommsPVBUS.bus_slave	PVBUSSlave

### Ports for GICv3CommsPVBUS

Port	Direction	Protocol	Description
axi_manager_id_s	slave	Value_64	-
distributor_s	slave	GICv3Comms	-
pvbus_m	master	PVBUS	-
pvbus_s	slave	PVBUS	-

### Parameters for GICv3CommsPVBUS

This component does not have any parameters.

## 3.203 GICv3ProtocolChecker

Defined in `LISA/GICv3ProtocolChecker.lisa`.

### About GICv3ProtocolChecker

GICv3 Component for command protocol checking.

### Iris and MTI instances for GICv3ProtocolChecker

This model has the following Iris instances:

Name	Instance type
GICv3ProtocolChecker	GICv3ProtocolChecker

This model has the following MTI trace components:

Name	Component type
GICv3ProtocolChecker	GICv3ProtocolChecker

### Ports for GICv3ProtocolChecker

Port	Direction	Protocol	Description
cpu_comms	master	GICv3Comms	Master GICv3Comms port.
gicv3_comms	slave	GICv3Comms	Slave GICv3Comms port.

### Parameters for GICv3ProtocolChecker

#### **cpu\_interface\_id**

Cpu interface id to which this component is connected.

Type: `uint8_t`

Default value: `0`

#### **enable\_protocol\_checking**

Enable/disable the protocol checking.

Type: `bool`

Default value: `true`

## 3.204 GICv5

Defined in `LISA/GICv5.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
N/A	Alpha support

For an explanation of the quality levels, see [Quality level definitions](#).

### About GICv5

GICv5 metacomponent for redistribution of interrupts (contains IWB and configurable numbers of ITSs and IRSs).

### Iris and MTI instances for GICv5

This model has the following Iris instances:

Name	Instance type
GICv5	GICv5
GICv5.irs0	IRS
GICv5.irs0.EL3	IRSDomain
GICv5.irs0.NON_SECURE	IRSDomain
GICv5.irs0.SECURE	IRSDomain
GICv5.its0	ITS
GICv5.its0.EL3	ITSDomain
GICv5.its0.NON_SECURE	ITSDomain
GICv5.its0.SECURE	ITSDomain
GICv5.iwb0	IWB
GICv5.table_read_m[0].pvbusmaster	PVBusMaster

This model has the following MTI trace components:

Name	Component type
GICv5	GICv5
GICv5.irs0	IRS
GICv5.irs0.EL3	IRSDomain
GICv5.irs0.NON_SECURE	IRSDomain
GICv5.irs0.SECURE	IRSDomain
GICv5.its0	ITS
GICv5.its0.EL3	ITSDomain
GICv5.its0.NON_SECURE	ITSDomain
GICv5.its0.SECURE	ITSDomain
GICv5.iwb0	IWB

Name	Component type
GICv5.table_read_m[0].pvbusmaster	PVBusMaster

## Ports for GICv5

Port	Direction	Protocol	Description
gicv5_irs0_pvbus_m	master	PVBus	-
gicv5_irs0_pvbus_s	slave	PVBus	-
gicv5_irs10_pvbus_m	master	PVBus	-
gicv5_irs10_pvbus_s	slave	PVBus	-
gicv5_irs11_pvbus_m	master	PVBus	-
gicv5_irs11_pvbus_s	slave	PVBus	-
gicv5_irs12_pvbus_m	master	PVBus	-
gicv5_irs12_pvbus_s	slave	PVBus	-
gicv5_irs13_pvbus_m	master	PVBus	-
gicv5_irs13_pvbus_s	slave	PVBus	-
gicv5_irs14_pvbus_m	master	PVBus	-
gicv5_irs14_pvbus_s	slave	PVBus	-
gicv5_irs15_pvbus_m	master	PVBus	-
gicv5_irs15_pvbus_s	slave	PVBus	-
gicv5_irs16_pvbus_m	master	PVBus	-
gicv5_irs16_pvbus_s	slave	PVBus	-
gicv5_irs17_pvbus_m	master	PVBus	-
gicv5_irs17_pvbus_s	slave	PVBus	-
gicv5_irs18_pvbus_m	master	PVBus	-
gicv5_irs18_pvbus_s	slave	PVBus	-
gicv5_irs19_pvbus_m	master	PVBus	-
gicv5_irs19_pvbus_s	slave	PVBus	-
gicv5_irs1_pvbus_m	master	PVBus	-
gicv5_irs1_pvbus_s	slave	PVBus	-
gicv5_irs20_pvbus_m	master	PVBus	-
gicv5_irs20_pvbus_s	slave	PVBus	-
gicv5_irs21_pvbus_m	master	PVBus	-
gicv5_irs21_pvbus_s	slave	PVBus	-
gicv5_irs22_pvbus_m	master	PVBus	-
gicv5_irs22_pvbus_s	slave	PVBus	-
gicv5_irs23_pvbus_m	master	PVBus	-
gicv5_irs23_pvbus_s	slave	PVBus	-
gicv5_irs24_pvbus_m	master	PVBus	-
gicv5_irs24_pvbus_s	slave	PVBus	-
gicv5_irs25_pvbus_m	master	PVBus	-
gicv5_irs25_pvbus_s	slave	PVBus	-

Port	Direction	Protocol	Description
gicv5_irs26_pvbuss_m	master	PVBus	-
gicv5_irs26_pvbuss_s	slave	PVBus	-
gicv5_irs27_pvbuss_m	master	PVBus	-
gicv5_irs27_pvbuss_s	slave	PVBus	-
gicv5_irs28_pvbuss_m	master	PVBus	-
gicv5_irs28_pvbuss_s	slave	PVBus	-
gicv5_irs29_pvbuss_m	master	PVBus	-
gicv5_irs29_pvbuss_s	slave	PVBus	-
gicv5_irs2_pvbuss_m	master	PVBus	-
gicv5_irs2_pvbuss_s	slave	PVBus	-
gicv5_irs30_pvbuss_m	master	PVBus	-
gicv5_irs30_pvbuss_s	slave	PVBus	-
gicv5_irs31_pvbuss_m	master	PVBus	-
gicv5_irs31_pvbuss_s	slave	PVBus	-
gicv5_irs3_pvbuss_m	master	PVBus	-
gicv5_irs3_pvbuss_s	slave	PVBus	-
gicv5_irs4_pvbuss_m	master	PVBus	-
gicv5_irs4_pvbuss_s	slave	PVBus	-
gicv5_irs5_pvbuss_m	master	PVBus	-
gicv5_irs5_pvbuss_s	slave	PVBus	-
gicv5_irs6_pvbuss_m	master	PVBus	-
gicv5_irs6_pvbuss_s	slave	PVBus	-
gicv5_irs7_pvbuss_m	master	PVBus	-
gicv5_irs7_pvbuss_s	slave	PVBus	-
gicv5_irs8_pvbuss_m	master	PVBus	-
gicv5_irs8_pvbuss_s	slave	PVBus	-
gicv5_irs9_pvbuss_m	master	PVBus	-
gicv5_irs9_pvbuss_s	slave	PVBus	-
gicv5_iwb0_wire_in	slave	Signal	-
gicv5_iwb10_wire_in	slave	Signal	-
gicv5_iwb11_wire_in	slave	Signal	-
gicv5_iwb12_wire_in	slave	Signal	-
gicv5_iwb13_wire_in	slave	Signal	-
gicv5_iwb14_wire_in	slave	Signal	-
gicv5_iwb15_wire_in	slave	Signal	-
gicv5_iwb16_wire_in	slave	Signal	-
gicv5_iwb17_wire_in	slave	Signal	-
gicv5_iwb18_wire_in	slave	Signal	-
gicv5_iwb19_wire_in	slave	Signal	-
gicv5_iwb1_wire_in	slave	Signal	-

Port	Direction	Protocol	Description
gicv5_iwb20_wire_in	slave	Signal	-
gicv5_iwb21_wire_in	slave	Signal	-
gicv5_iwb22_wire_in	slave	Signal	-
gicv5_iwb23_wire_in	slave	Signal	-
gicv5_iwb24_wire_in	slave	Signal	-
gicv5_iwb25_wire_in	slave	Signal	-
gicv5_iwb26_wire_in	slave	Signal	-
gicv5_iwb27_wire_in	slave	Signal	-
gicv5_iwb28_wire_in	slave	Signal	-
gicv5_iwb29_wire_in	slave	Signal	-
gicv5_iwb2_wire_in	slave	Signal	-
gicv5_iwb30_wire_in	slave	Signal	-
gicv5_iwb31_wire_in	slave	Signal	-
gicv5_iwb3_wire_in	slave	Signal	-
gicv5_iwb4_wire_in	slave	Signal	-
gicv5_iwb5_wire_in	slave	Signal	-
gicv5_iwb6_wire_in	slave	Signal	-
gicv5_iwb7_wire_in	slave	Signal	-
gicv5_iwb8_wire_in	slave	Signal	-
gicv5_iwb9_wire_in	slave	Signal	-
gicv5_spi_wire_in	slave	Signal	-
po_reset	slave	Signal	-
pvbush_filtermiss_m	master	PVBus	Passthrough for accesses to pages not used by the GIC IRI.
pvbush_m	master	PVBus	Memory bus out: transactions generated by the GICv5.
pvbush_s	slave	PVBus	Memory bus in: memory-mapped register accesses are accepted on this interface.
reset	slave	Signal	Resets.
wake_request	master	Signal	Power management outputs.

## Parameters for GICv5

### **config\_file**

File path for the GICv5 configuration yaml. The file list the GICv5 params.

Type: string

Default value: N/A

### **enabled**

If set, then GICv5 is enabled.

Type: bool

Default value: `true`

### **has\_gcie\_legacy**

When set to true, FEAT\_GCIE\_LEGACY is supported.

Type: `bool`

Default value: `false`

### **has\_rme**

Type: `bool`

Default value: `false`

### **num\_cores**

Number of cores implemented.

Type: `uint32_t`

Default value: `0x1`

### **output\_attributes**

Encodes the transform the GIC applies to encode various attributes on the bus.

Type: `string`

Default value: `"ExtendedID[62:55]=MPAM_PMG, ExtendedID[54:39]=MPAM_PARTID, ExtendedID[38]=MPAM_NS, UserFlags[31:16]=MECID"`

### **rme\_default\_mecid\_nonsecure**

MECID to use for Non Secure accesses.

Type: `uint16`

Default value: `0`

### **rme\_default\_mecid\_realm**

Default MECID to use for Realm accesses.

Type: `uint16`

Default value: `0`

### **rme\_default\_mecid\_root**

MECID to use for Root accesses.

Type: `uint16`



Default value: 0

**rme\_default\_mecid\_secure**

MECID to use for Secure accesses.

Type: uint16

Default value: 0

**support\_virtualization**

If set, then GICv5 supports virtualization.

Type: bool

Default value: false

3.205 GUIPoll

Defined in LISA/GUIPoll.lisa.

About GUIPoll

An external subcomponent that encapsulates support for generating a real-time callback signal that can be used to poll the event queue of a visualisation GUI.

The `gui_callback()` method of the callback port is invoked periodically, at approximately the rate determined by the `delay_ms` parameter.



Note

This callback is real-time, not simulation-time. Also, callbacks will continue even while the simulation is paused. Because of this, the client code should not implement a callback behavior that can modify the state of the simulation.

Ports for GUIPoll

Port	Direction	Protocol	Description
gui_callback	master	GUIPollCallback	Sends callback requests to the visualization component.

Parameters for GUIPoll

**delay\_ms**

Determines the period, in milliseconds of real time, between `gui_callback()` calls.

Type: uint32\_t

Default value: 50

**has\_gui**

Set to false if GUI is disabled.

Type: bool

Default value: true

## 3.206 Generic\_PLL

Defined in LISA/Generic\_PLL.lisa.

### About Generic\_PLL

PLL class for Client system.

### Iris and MTI instances for Generic\_PLL

This model has the following Iris instances:

Name	Instance type
Generic_PLL	Generic_PLL
Generic_PLL.clockWatcher	FrequencyProbe
Generic_PLL.divider	ClockDivider
Generic_PLL.vco_divider	ClockDivider

This model has the following MTI trace components:

Name	Component type
Generic_PLL.divider	ClockDivider
Generic_PLL.vco_divider	ClockDivider

### Ports for Generic\_PLL

Port	Direction	Protocol	Description
bypass	slave	Signal	-
clk_in	slave	ClockSignal	-
clk_out	master	ClockSignal	-
div_enable	slave	Signal	-
fbdiv	slave	ValueState	-
frac	slave	ValueState	-
lock	master	Signal	-
pll_enable	slave	Signal	-
postdiv1	slave	ValueState	-
postdiv2	slave	ValueState	-
refdiv	slave	ValueState	-

Port	Direction	Protocol	Description
vco_clk_out	master	ClockSignal	-
vco_enable	slave	Signal	-

## Parameters for Generic\_PLL

### diagnostics

Diagnostics.

Type: uint8\_t

Default value: 2

## 3.207 HostBridge

Defined in LISA/HostBridge.lisa.

### About HostBridge

This component acts as a networking proxy for target NIC device models, to forward and receive ethernet packets to and from the host. Two kinds of proxy backend are integrated into this component:

- A host TAP/TUN-like network device, which is an ordinary TAP or MacVTap. This is the default.
- User-mode networking, which emulates a built-in IP router and DHCP server to route traffic using the host user-mode socket layer. To enable user-mode networking, set the `userNetworking` parameter to true.

HostBridge requires the following initialization sequence:

```
hostbridge.state.setValue(HostBridge::STATUS);
hostbridge.state.setValue(HostBridge::S_UP);
```

To enable tracing of user-mode networking, which can help to debug networking issues, set the `FASTSIM_USERNET_DUMP` environment variable to any or all of the following values:

```
arpin, arpout, udpin, udpout, etherin, etherout, ipv4in, ipv4out,
ipv4fragin, ipv4fragout, tcpin, tcpout, dhcpv4in, dhcpv4out
```

### See also

- [Configuring the networking environment for Linux](#)
- [User mode networking](#)

### Iris and MTI instances for HostBridge

This model has the following Iris instances:

Name	Instance type
HostBridge	HostBridge

### Ports for HostBridge

Port	Direction	Protocol	Description
eth	slave	VirtualEthernet	-
state	slave	ValueState_64	-

### Parameters for HostBridge

#### **interfaceName**

Host Interface.

Type: `string`

Default value: `""`

#### **userNetOptions**

Control options for UserNet TCP/IP (for internal use only, please do not use).

Type: `string`

Default value: `""`

#### **userNetPorts**

Listening ports to expose in user-mode networking.

Type: `string`

Default value: `""`

#### **userNetSubnet**

Virtual subnet for user-mode networking.

Type: `string`

Default value: `"172.20.51.0/24"`

#### **userNetworking**

Enable user-mode networking.

Type: `bool`

Default value: `false`

## 3.208 HostSerialInterface

Defined in `LISA/HostSerialInterface.lisa`.

### About HostSerialInterface

Component which provides access to the host serial interface.

### Iris and MTI instances for HostSerialInterface

This model has the following Iris instances:

Name	Instance type
HostSerialInterface	<a href="#">HostSerialInterface</a>

### Ports for HostSerialInterface

Port	Direction	Protocol	Description
SerialData	slave	<a href="#">SerialData</a>	Serial data connection to export to host machine.

### Parameters for HostSerialInterface

#### **baud\_rate**

Baud rate override.

Type: `int`

Default value: 0

#### **device**

HW device to use.

Type: `string`

Default value: `"/dev/ttyS0"`

## 3.209 ICS307

Defined in `LISA/ICS307.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

## About ICS307

Use this component to convert the rate of one ClockSignal to another ClockSignal by using configurable multiplier, divider, and scale values. The divider ratio can be set by startup parameters or at runtime by a configuration port. Changes to the input ClockSignal rate and divider ratio are reflected immediately by the output ClockSignal ports.

Three values determine the divisor ratio:

- `vdw`
- `rdw`
- `od`

To calculate the divisor ratio, use:

$$\text{Divisor} = ((\text{rdw}+2) * \text{scale}) / (2 * (\text{vdw}+8))$$

where `scale` is derived from this table indexed by `od`:

**Table 3-730: od to scale conversion**

od	scale
0	10
1	2
2	8
3	4
4	5
5	7
6	3
7	6

The default values of `vdw`, `rdw` and `od` are 4, 6, and 3 to give a default divisor rate of:

$$((6+2) * 4) / (2 * (4+8)) = 4/3$$

## Iris and MTI instances for ICS307

This model has the following Iris instances:

Name	Instance type
ICS307	ICS307
ICS307.clkdiv_clk1	ClockDivider

This model has the following MTI trace components:

Name	Component type
ICS307.clkdiv_clk1	ClockDivider

## Ports for ICS307

Port	Direction	Protocol	Description
clk_in	slave	<a href="#">ClockSignal</a>	Master clock rate.
clk_out_clk1	master	<a href="#">ClockSignal</a>	Modified clock rate.
clk_out_ref	master	<a href="#">ClockSignal</a>	Pass through of master clock rate for divider chaining.
configuration	slave	<a href="#">ICS307Configuration</a>	Configuration port for setting divider ratio dynamically.

## Parameters for ICS307

### **od**

OD.

Type: `uint32_t`

Default value: 3

### **rdr**

RDR.

Type: `uint32_t`

Default value: 6

### **vdw**

VDW.

Type: `uint32_t`

Default value: 4

## 3.210 IDAU

Defined in `LISA/IDAU.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### IDAU interface

The Implementation Defined Attribution Unit (IDAU) model uses the `pv::IDAUsignal` struct to return the Security attributes for the address passed to it.

Unlike the hardware, the CPU Fast Model does not query the IDAU for each access. Communication is at a higher abstraction level to maintain simulation speed.

IDAU interface input signals:

**IDAUADDR**

Address of the region

IDAU interface output signals:

**IDAUNS**

Non-secure region response

**IDAUNSC**

Non-secure-callable region response

**IDAUID**

Region number

**IDAUIDV**

Region number valid

**IDAUNCHK**

Region exempt from attribution check

## IDAU transaction-level communication protocol

IDAU has the following ports:

- **slave port<PVBUS> pvbus\_s;**

PVBUS memory-based slave port. Masters can read or write to this port as follows:

- **Read**

Read returns IDAU region's `pv::IdauRegion` struct (32 byte), containing information about the IDAU region for the requested address. `pv::IdauRegion` contains the start address, end address, `pv::IDAUSignal`, and 8 bytes of padding (to make it 32 byte-aligned).

- **Write**

This port only supports 32 byte Write operations to pass in an `pv::IdauRegion` struct for updating an internal IDAU region.

- **DMI**

This port adds support for DMI requests and provides a pointer to a `pv::IdauRegion` for the requested address. An 'invalid DMI' call back occurs if the IDAU updates its regions.

- **master port<Value\_64> invalidate\_region;**

This port is used as a call back to inform masters that the IDAU has updated its region information.





To disable the IDAU, set the `NUM_IDAU_REGION` parameter to zero.

## Iris and MTI instances for IDAU

This model has the following Iris instances:

Name	Instance type
<code>IDAU.bus_bridge</code>	<code>PVBusBridge</code>

## Ports for IDAU

Port	Direction	Protocol	Description
<code>invalidate_region</code>	master	<code>Value_64</code>	This port is used as a call back to inform masters that the IDAU has updated its region information.
<code>pvbus_s</code>	slave	<code>PVBus</code>	-

## Parameters for IDAU

### **IDAU\_REGION0.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: `0`

### **IDAU\_REGION0.ENABLE**

Controls if region is S or NS, only valid when `NSC=0`. If `NSC=1` this parameter is ignored. 1 = region is NS, 0 = region is S (absent if `LADDR=0`).

Type: `bool`

Default value: `false`

### **IDAU\_REGION0.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

### **IDAU\_REGION0.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION0.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION1.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION1.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION1.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION1.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION1.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION10.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION10.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION10.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION10.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION10.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION100.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION100.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION100.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION100.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION100.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION101.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION101.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION101.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION101.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION101.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION102.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION102.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION102.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION102.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION102.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION103.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION103.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION103.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION103.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION103.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION104.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION104.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION104.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION104.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION104.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION105.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION105.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION105.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION105.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION105.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION106.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION106.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION106.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION106.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION106.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION107.BADDR**

Base address of IDAU region.



Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION107.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION107.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION107.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION107.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION108.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION108.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION108.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION108.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION108.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION109.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION109.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION109.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION109.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION109.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION11.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION11.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION11.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION11.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION11.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION110.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION110.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION110.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION110.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION110.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION111.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION111.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION111.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION111.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION111.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION112.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION112.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION112.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION112.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION112.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION113.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION113.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION113.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION113.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION113.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION114.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION114.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION114.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION114.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION114.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION115.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION115.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION115.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION115.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION115.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION116.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION116.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION116.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION116.LADDR**

Limit address of IDAU region.

Type: `uint32_t`



Default value: 0

**IDAU\_REGION116.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION117.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION117.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION117.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION117.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION117.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION118.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION118.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION118.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION118.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION118.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION119.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION119.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION119.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION119.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION119.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION12.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION12.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION12.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION12.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION12.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION120.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION120.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION120.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION120.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION120.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION121.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION121.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION121.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION121.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION121.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION122.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION122.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION122.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION122.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION122.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION123.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION123.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION123.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION123.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION123.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION124.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION124.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION124.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION124.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION124.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION125.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION125.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION125.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION125.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION125.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION126.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION126.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`



**IDAU\_REGION126.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION126.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION126.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION127.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION127.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION127.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION127.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION127.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION128.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION128.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION128.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION128.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION128.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION129.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: `0`

#### **IDAU\_REGION129.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION129.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION129.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: `0`

#### **IDAU\_REGION129.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION13.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: `0`

#### **IDAU\_REGION13.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION13.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION13.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION13.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION130.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION130.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION130.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION130.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION130.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION131.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION131.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION131.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION131.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION131.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION132.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: `0`

#### **IDAU\_REGION132.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION132.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION132.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: `0`

#### **IDAU\_REGION132.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION133.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: `0`

#### **IDAU\_REGION133.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION133.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION133.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION133.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION134.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION134.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION134.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION134.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION134.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION135.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION135.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION135.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION135.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION135.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION136.BADDR**

Base address of IDAU region.



Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION136.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION136.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION136.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION136.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION137.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION137.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION137.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION137.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION137.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION138.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION138.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION138.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION138.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION138.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION139.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION139.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION139.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION139.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION139.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION14.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION14.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION14.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION14.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION14.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION140.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION140.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION140.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION140.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION140.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION141.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION141.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION141.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION141.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION141.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION142.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION142.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION142.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION142.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION142.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION143.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION143.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION143.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION143.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION143.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION144.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION144.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION144.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION144.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: `0`

**IDAU\_REGION144.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION145.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: `0`

**IDAU\_REGION145.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION145.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION145.LADDR**

Limit address of IDAU region.

Type: `uint32_t`



Default value: 0

**IDAU\_REGION145.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION146.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION146.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION146.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION146.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION146.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION147.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION147.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION147.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION147.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION147.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION148.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION148.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION148.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION148.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION148.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION149.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION149.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION149.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION149.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION149.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION15.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION15.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION15.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION15.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION15.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION150.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION150.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION150.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION150.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION150.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION151.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION151.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION151.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION151.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION151.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION152.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION152.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION152.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION152.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION152.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION153.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION153.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION153.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION153.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION153.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION154.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION154.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION154.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION154.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION154.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION155.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION155.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`



**IDAU\_REGION155.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION155.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION155.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION156.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION156.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION156.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION156.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION156.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION157.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION157.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION157.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION157.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION157.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION158.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: `0`

#### **IDAU\_REGION158.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION158.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION158.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: `0`

#### **IDAU\_REGION158.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION159.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: `0`

#### **IDAU\_REGION159.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION159.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION159.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION159.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION16.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION16.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION16.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION16.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION16.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION160.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION160.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION160.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION160.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION160.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION161.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION161.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION161.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION161.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION161.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION162.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION162.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION162.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION162.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION162.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION163.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION163.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION163.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION163.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION163.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION164.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION164.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION164.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION164.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION164.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION165.BADDR**

Base address of IDAU region.



Type: `uint32_t`

Default value: `0`

#### **IDAU\_REGION165.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION165.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION165.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: `0`

#### **IDAU\_REGION165.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION166.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: `0`

#### **IDAU\_REGION166.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION166.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION166.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION166.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION167.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION167.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION167.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION167.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION167.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION168.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION168.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION168.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION168.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION168.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION169.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION169.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION169.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION169.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION169.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION17.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION17.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION17.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION17.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION17.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION170.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION170.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION170.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION170.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION170.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION171.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION171.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION171.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION171.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION171.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION172.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: `0`

#### **IDAU\_REGION172.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION172.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION172.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: `0`

#### **IDAU\_REGION172.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION173.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: `0`

#### **IDAU\_REGION173.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION173.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION173.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION173.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION174.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION174.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION174.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION174.LADDR**

Limit address of IDAU region.

Type: `uint32_t`



Default value: 0

**IDAU\_REGION174.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION175.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION175.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION175.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION175.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION175.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION176.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION176.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION176.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION176.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION176.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION177.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION177.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION177.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION177.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION177.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION178.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION178.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION178.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION178.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION178.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION179.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION179.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION179.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION179.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION179.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION18.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: `0`

#### **IDAU\_REGION18.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION18.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION18.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: `0`

#### **IDAU\_REGION18.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION180.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: `0`

#### **IDAU\_REGION180.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION180.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION180.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION180.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION181.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION181.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION181.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION181.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION181.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION182.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION182.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION182.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION182.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION182.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION183.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: `0`

#### **IDAU\_REGION183.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION183.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION183.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: `0`

#### **IDAU\_REGION183.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION184.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: `0`

#### **IDAU\_REGION184.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`



**IDAU\_REGION184.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION184.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION184.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION185.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION185.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION185.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION185.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION185.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION186.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION186.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION186.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION186.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION186.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION187.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION187.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION187.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION187.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION187.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION188.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION188.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION188.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION188.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION188.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION189.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION189.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION189.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION189.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION189.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION19.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION19.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION19.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION19.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION19.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION190.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION190.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION190.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION190.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION190.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION191.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION191.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION191.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION191.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION191.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION192.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION192.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION192.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION192.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION192.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION193.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION193.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION193.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION193.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION193.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION194.BADDR**

Base address of IDAU region.



Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION194.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION194.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION194.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION194.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION195.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION195.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION195.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION195.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION195.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION196.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION196.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION196.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION196.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION196.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION197.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION197.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION197.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION197.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION197.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION198.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION198.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION198.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION198.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION198.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION199.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION199.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION199.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION199.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION199.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION2.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION2.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION2.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION2.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION2.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION20.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION20.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION20.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION20.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION20.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION200.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION200.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION200.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION200.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION200.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION201.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION201.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION201.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION201.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION201.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION202.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION202.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION202.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION202.LADDR**

Limit address of IDAU region.

Type: `uint32_t`



Default value: 0

**IDAU\_REGION202.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION203.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION203.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION203.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION203.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION203.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION204.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION204.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION204.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION204.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION204.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION205.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION205.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION205.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION205.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION205.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION206.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION206.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION206.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION206.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION206.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION207.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION207.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION207.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION207.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION207.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION208.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION208.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION208.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION208.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION208.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION209.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION209.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION209.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION209.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION209.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION21.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION21.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION21.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION21.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION21.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION210.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION210.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION210.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION210.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION210.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION211.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION211.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION211.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION211.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION211.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION212.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION212.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`



**IDAU\_REGION212.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION212.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION212.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION213.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION213.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION213.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION213.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION213.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION214.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION214.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION214.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION214.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION214.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION215.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: `0`

#### **IDAU\_REGION215.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION215.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION215.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: `0`

#### **IDAU\_REGION215.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION216.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: `0`

#### **IDAU\_REGION216.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION216.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION216.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION216.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION217.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION217.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION217.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION217.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION217.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION218.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION218.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION218.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION218.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION218.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION219.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION219.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION219.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION219.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION219.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION22.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION22.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION22.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION22.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION22.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION220.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION220.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION220.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION220.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION220.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION221.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION221.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION221.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION221.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION221.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION222.BADDR**

Base address of IDAU region.



Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION222.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION222.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION222.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION222.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION223.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION223.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION223.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION223.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION223.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION224.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION224.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION224.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION224.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION224.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION225.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION225.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION225.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION225.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION225.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION226.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION226.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION226.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION226.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION226.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION227.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION227.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION227.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION227.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION227.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION228.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION228.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION228.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION228.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION228.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION229.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION229.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION229.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION229.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION229.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION23.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION23.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION23.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION23.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION23.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION230.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION230.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION230.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION230.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION230.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION231.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION231.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION231.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION231.LADDR**

Limit address of IDAU region.

Type: `uint32_t`



Default value: 0

**IDAU\_REGION231.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION232.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION232.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION232.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION232.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION232.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION233.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION233.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION233.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION233.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION233.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION234.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION234.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION234.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION234.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION234.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION235.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION235.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION235.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION235.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION235.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION236.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION236.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION236.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION236.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION236.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION237.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION237.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION237.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION237.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION237.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION238.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION238.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION238.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION238.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION238.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION239.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION239.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION239.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION239.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION239.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION24.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION24.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION24.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION24.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION24.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION240.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION240.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION240.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION240.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION240.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION241.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION241.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`



**IDAU\_REGION241.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION241.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION241.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION242.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION242.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION242.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION242.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION242.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION243.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION243.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION243.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION243.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION243.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION244.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION244.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION244.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION244.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION244.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION245.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION245.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION245.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION245.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION245.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION246.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION246.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION246.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION246.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION246.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION247.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION247.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION247.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION247.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION247.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION248.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION248.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION248.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION248.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION248.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION249.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION249.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION249.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION249.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION249.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION25.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION25.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION25.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION25.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION25.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION250.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION250.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION250.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION250.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION250.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION251.BADDR**

Base address of IDAU region.



Type: `uint32_t`

Default value: `0`

#### **IDAU\_REGION251.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION251.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION251.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: `0`

#### **IDAU\_REGION251.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION252.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: `0`

#### **IDAU\_REGION252.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION252.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION252.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION252.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION253.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION253.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION253.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION253.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION253.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION254.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION254.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION254.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION254.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION254.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION255.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION255.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION255.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION255.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION255.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION26.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION26.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION26.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION26.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION26.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION27.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION27.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION27.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION27.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION27.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION28.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION28.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION28.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION28.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION28.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION29.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION29.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION29.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION29.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION29.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION3.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION3.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION3.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION3.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION3.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION30.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION30.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION30.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION30.LADDR**

Limit address of IDAU region.

Type: `uint32_t`



Default value: 0

**IDAU\_REGION30.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION31.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION31.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION31.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION31.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION31.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION32.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: `0`

#### **IDAU\_REGION32.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION32.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION32.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: `0`

#### **IDAU\_REGION32.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION33.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: `0`

#### **IDAU\_REGION33.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION33.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION33.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION33.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION34.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION34.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION34.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION34.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION34.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION35.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION35.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION35.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION35.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION35.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION36.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION36.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION36.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION36.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION36.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION37.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION37.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION37.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION37.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION37.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION38.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION38.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION38.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION38.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION38.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION39.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION39.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION39.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION39.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION39.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION4.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION4.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION4.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION4.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION4.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION40.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION40.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`



**IDAU\_REGION40.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION40.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION40.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION41.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION41.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION41.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION41.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION41.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION42.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION42.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION42.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION42.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION42.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION43.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: `0`

#### **IDAU\_REGION43.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION43.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION43.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: `0`

#### **IDAU\_REGION43.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION44.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: `0`

#### **IDAU\_REGION44.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION44.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION44.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION44.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION45.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION45.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION45.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION45.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION45.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION46.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION46.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION46.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION46.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION46.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION47.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION47.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION47.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION47.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION47.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION48.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION48.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION48.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION48.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION48.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION49.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION49.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION49.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION49.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION49.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION5.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION5.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION5.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION5.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION5.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION50.BADDR**

Base address of IDAU region.



Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION50.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION50.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION50.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION50.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION51.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION51.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION51.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION51.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION51.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION52.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION52.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION52.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION52.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION52.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION53.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION53.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION53.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION53.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION53.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION54.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION54.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION54.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION54.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION54.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION55.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION55.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION55.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION55.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION55.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION56.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION56.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION56.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION56.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION56.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION57.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION57.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION57.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION57.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION57.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION58.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION58.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION58.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION58.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION58.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION59.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION59.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION59.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION59.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION59.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION6.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION6.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION6.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION6.LADDR**

Limit address of IDAU region.

Type: `uint32_t`



Default value: 0

**IDAU\_REGION6.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION60.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION60.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION60.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION60.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION60.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION61.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION61.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION61.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION61.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION61.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION62.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION62.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION62.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION62.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION62.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION63.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION63.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION63.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION63.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION63.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION64.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION64.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION64.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION64.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION64.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION65.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION65.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION65.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION65.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION65.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION66.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION66.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION66.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION66.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION66.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION67.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION67.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION67.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION67.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION67.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION68.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION68.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION68.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION68.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION68.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION69.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION69.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION69.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION69.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION69.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION7.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION7.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`



**IDAU\_REGION7.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION7.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION7.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION70.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION70.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION70.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION70.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION70.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION71.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION71.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION71.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION71.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION71.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION72.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION72.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION72.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION72.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION72.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION73.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION73.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION73.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION73.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION73.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION74.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION74.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION74.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION74.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION74.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION75.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION75.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION75.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION75.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION75.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION76.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION76.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION76.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION76.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION76.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION77.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION77.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION77.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION77.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION77.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION78.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION78.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION78.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION78.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION78.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION79.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION79.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION79.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION79.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION79.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION8.BADDR**

Base address of IDAU region.



Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION8.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION8.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION8.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION8.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION80.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION80.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION80.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION80.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION80.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION81.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION81.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION81.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION81.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION81.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION82.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION82.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION82.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION82.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION82.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION83.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: `0`

#### **IDAU\_REGION83.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION83.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION83.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: `0`

#### **IDAU\_REGION83.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION84.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: `0`

#### **IDAU\_REGION84.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION84.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION84.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION84.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION85.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION85.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION85.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION85.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION85.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION86.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION86.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION86.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION86.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION86.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION87.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: `0`

#### **IDAU\_REGION87.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION87.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION87.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: `0`

#### **IDAU\_REGION87.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION88.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: `0`

#### **IDAU\_REGION88.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION88.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION88.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION88.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION89.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION89.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION89.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION89.LADDR**

Limit address of IDAU region.

Type: `uint32_t`



Default value: 0

**IDAU\_REGION89.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION9.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION9.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION9.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION9.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION9.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION90.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION90.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION90.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION90.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION90.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION91.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION91.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION91.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION91.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION91.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION92.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION92.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION92.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION92.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION92.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION93.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION93.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION93.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION93.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION93.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION94.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION94.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION94.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION94.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION94.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION95.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION95.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION95.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION95.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION95.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION96.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION96.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION96.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION96.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION96.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION97.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION97.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

**IDAU\_REGION97.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION97.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

**IDAU\_REGION97.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**IDAU\_REGION98.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION98.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`

#### **IDAU\_REGION98.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION98.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION98.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

#### **IDAU\_REGION99.BADDR**

Base address of IDAU region.

Type: `uint32_t`

Default value: 0

#### **IDAU\_REGION99.ENABLE**

Controls if region is S or NS, only valid when NSC=0. If NSC=1 this parameter is ignored. 1 = region is NS, 0 = region is S (absent if LADDR=0).

Type: `bool`

Default value: `false`



**IDAU\_REGION99.EXEMPT**

Mark IDAU region as exempt.

Type: `bool`

Default value: `false`

**IDAU\_REGION99.LADDR**

Limit address of IDAU region.

Type: `uint32_t`

Default value: `0`

**IDAU\_REGION99.NSC**

Controls if region is NSC. 1 = IDAU region is NSC, 0 = ENABLE parameter determines if S or NS.

Type: `bool`

Default value: `false`

**NUM\_IDAU\_REGION**

Type: `uint32_t`

Default value: `0`

## 3.211 ILCU

Defined in `LISA/ILCU.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following parameters were added:

- `rse_1_6`

### About ILCU

Integraton Layer Control Unit.

## Iris and MTI instances for ILCU

This model has the following Iris instances:

Name	Instance type
ILCU	ILCU

## Ports for ILCU

Port	Direction	Protocol	Description
apb_control_in	slave	PVBus	APB3 Subordinate Interface - Access to ILCU registers
clk_div_control_out	master	Value	3 bits to control the Cryptocell clock divider
crypto_irq_in	slave	Signal	Interrupt input driven by the Cryptocell
crypto_irq_out	master	Signal	Interrupt output destined for the processor
ic_trigger_in_ack_out	master	Signal	Integrity checker trigger acknowledge signal
ic_trigger_in_req_in	slave	Signal	Integrity checker trigger request signal
irq_mux_control_out	master	Signal	IRQ Mux control bit that internally selects where crypto_irq_in is routed
lcm_seed_lfsr_data_out	master	Value_64	64 bit seed value for the LFSR interface
lcm_seed_lfsr_valid_out	master	Signal	Indicates that the lcm_seed_lfsr_data signal is valid
lcs_in	slave	Value	3 bits giving the lifecycle state from the RSE persistent state interface (PSI)
lcs_valid_in	slave	Signal	Signal from the RSE PSI indicating whether the lifecycle state is valid
n_coldreset_in	slave	Signal	ICLU reset in
otpw_otp_is_ready_in	slave	Signal	Status signal from the OTP wrapper indicating it is ready for commands
secure_gpo_out	master	Value	16 bits of general purpose output (8 bits programmed + 8 complemented bits driven)
trigger_mux_control_out	master	Value	2 bits input to the DMA Trigger Mux to internally select the DMA trigger source
trigger_out_ack_in	slave	Signal	DMA Trigger from one of the muxed sources acknowledge signal
trigger_out_req_out	master	Signal	DMA Trigger from one of the muxed sources request signal
trng_trigger_out	master	Signal	DMA Trigger Mux input from the IRQ Mux indicating whether the TRNG complete irq has occurred

## Parameters for ILCU

### LCM\_RNG\_SEED\_NOT\_REQUIRED

Configuration parameter LCM\_RNG\_SEED\_NOT\_REQUIRED: 0 ~ for systems where unpredictable delays between reads are used as a countermeasure. 1 ~ for systems which do not use a countermeasure with random delays.

Type: bool

Default value: 0

### diagnostics

Diagnostics.

Type: `uint32_t`

Default value: 0

### **rse\_1\_6**

Enable RSE 1.6 functionality.

Type: `bool`

Default value: 0

## 3.212 InstructionCount2SystemC

Defined in `examples/SystemCExport/Bridges/InstructionCount2SystemC.lisa`.



Note

Variants of this component also exist with multiple input and output ports.

### Iris and MTI instances for InstructionCount2SystemC

This model has the following Iris instances:

Name	Instance type
<code>InstructionCount2SystemC</code>	<code>InstructionCount2SystemC</code>

### Ports for InstructionCount2SystemC

Port	Direction	Protocol	Description
<code>inst_count</code>	slave	<a href="#">AMBAPVValueState64</a>	-
<code>run_state</code>	slave	<a href="#">AMBAPVValueState</a>	-
<code>ticks</code>	slave	<a href="#">InstructionCount</a>	-

### Parameters for InstructionCount2SystemC

This component does not have any parameters.

## 3.213 InstructionCount2SystemCx4

Defined in `examples/SystemCExport/Bridges/InstructionCount2SystemCx4.lisa`.

### About InstructionCount2SystemCx4

InstructionCount to SystemC Converter x4.

## Iris and MTI instances for InstructionCount2SystemCx4

This model has the following Iris instances:

Name	Instance type
InstructionCount2SystemCx4	InstructionCount2SystemCx4

## Ports for InstructionCount2SystemCx4

Port	Direction	Protocol	Description
inst_count	slave	AMBAPVValueState64	-
run_state	slave	AMBAPVValueState	-
ticks	slave	InstructionCount	-

## Parameters for InstructionCount2SystemCx4

This component does not have any parameters.

# 3.214 IntegrityChecker

Defined in `LISA/IntegrityChecker.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
0.5	Alpha support

For an explanation of the quality levels, see [Quality level definitions](#).

## About IntegrityChecker

Integrity Checker.

## Iris and MTI instances for IntegrityChecker

This model has the following Iris instances:

Name	Instance type
IntegrityChecker	IntegrityChecker
IntegrityChecker.Internal Master[0].pvbusmaster	PVBusMaster
IntegrityChecker.apb	PVBusSlave

This model has the following MTI trace components:

Name	Component type
IntegrityChecker.Internal Master[0].pvbusmaster	PVBusMaster
IntegrityChecker.apb	PVBusSlave

## Ports for IntegrityChecker

Port	Direction	Protocol	Description
apb	slave	PVBus	APB Subordinate Interface - Access to registers
ic_alarm_out	master	Signal	Alarm status out signal
ic_interrupt_out	master	Signal	Interrupt out signal
ic_match_trigger_ack_in	slave	Signal	IC Match done ack signal
ic_match_trigger_req_out	master	Signal	IC Match done req signal
pvbus_m	master	PVBus	To read and write to external memory
reset_in	slave	Signal	Reset in signal
warm_reset_in	slave	Signal	Warm Reset in signal

## Parameters for IntegrityChecker

### ICBC\_RESET\_VALUE

ICBC register reset value.

Type: uint32\_t

Default value: 0x11B

### ICDL\_CHUNK\_SIZE

ICDL\_CHUNK\_SIZE.

Type: uint8\_t

Default value: 8

### PID0\_RESET\_VALUE

PID 0 register reset value.

Type: uint32\_t

Default value: 0x0

### PID1\_RESET\_VALUE

PID 1 register reset value.

Type: uint32\_t

Default value: 0xB0

### diagnostics

Diagnostics.

Type: uint32\_t

Default value: 2

## 3.215 IntelStrataFlashJ3

Defined in `LISA/IntelStrataFlashJ3.lisa`.

### Changes in 11.31.15

The following parameters were added:

- `layout`

### About IntelStrataFlashJ3

This component is an efficient implementation of a NOR flash memory type device, an Intel StrataFlash Memory (J3). For information about Intel StrataFlash Memory (J3), see [Intel Download Center, Intel StrataFlash Memory \(J3\) datasheet](#).

In normal usage, the device acts as Read Only Memory (ROM) whose contents can be determined either by programming using the flashloader port or by using standard flash programming software running on the model, such as the Arm Firmware Suite.

The implementation of this component is approximately that of the Intel part in the VE development board. The component is effectively organized as a bank of two 16-bit Intel Flash components forming a 32-bit component that can be read or programmed in parallel. The component supports all hardware behavior except for:

- Protection register.
- Enhanced configuration register.
- Unique device identifier.
- One time programmable cells.
- Suspend/resume, which is silently ignored.
- Status interrupt line.

All block operations are atomic. This means that the status register state machine status bit always reads 1, ready.

In normal operation, this component has no user-visible registers, but you can read from it as if it is memory.

Programming it or changing the configuration requires a sequence of special write operations, see general flash programming documentation. The component supports Common Flash Interface query operations, which allow drivers to determine the properties of the flash memory.



The model interprets all writes as requests to the programming state machine, and there are many state-machine states that do not support subsequent reads and return `0xdeaddead` for them. Therefore, when simulating a ROM, use the `trapwrite=true` option.

## Setting the diagnostic level

Use the `diagnostics` parameter to select the level of diagnostic output:

### Level 0

None.

### Level 1

Report probable driver error operations:

- Unaligned operations that fault.
- Accesses that the state machine does not expect.
- Transitions of the state machine to unknown states.
- Writes to locked blocks and illegal lock commands.

### Level 2

Report unimplemented and therefore ignored operations, and log lock commands.

### Level 3

Warn if a flash write attempts to set bits. The write works if `unphysical_writes=true`.

### Level 4

Log every read and write.

## Defining the flash layout

Use the `layout` parameter to define the block structure of the flash device with more flexibility.

A flash device can include multiple erase regions. Each region can contain between 1 and 65536 blocks. All blocks within a region are identically sized, but block size can vary between regions. Block size must be a multiple of 256 bytes, ranging from 1 x 256 bytes to `0xFFFF` x 256 bytes. Regions, and blocks within them, are all contiguous.

For example:

```
layout="10@4096,128@0x40000,2@256"
```

This defines a layout with the following 3 regions:

- A region with 10 blocks of 4 KB each
- Followed by a region with 128 blocks of 256 KB each
- Followed by a region with 2 blocks of 256 bytes each

If both `layout` and `size` parameters are present, `layout` takes precedence.

## Iris and MTI instances for IntelStrataFlashJ3

This model has the following Iris instances:

Name	Instance type
IntelStrataFlashJ3	IntelStrataFlashJ3
IntelStrataFlashJ3.map	PVBusMapper
IntelStrataFlashJ3.mbs	PVBusSlave
IntelStrataFlashJ3.rmbs	PVBusSlave

This model has the following MTI trace components:

Name	Component type
IntelStrataFlashJ3.map	PVBusMapper
IntelStrataFlashJ3.mbs	PVBusSlave
IntelStrataFlashJ3.rmbs	PVBusSlave

## Ports for IntelStrataFlashJ3

Port	Direction	Protocol	Description
flashloader	slave	FlashLoaderPort	-
mem_port	slave	PVDevice	-
pvbus	slave	PVBus	-

## Parameters for IntelStrataFlashJ3

### diagnostics

Diagnostic level.

Type: uint32\_t

Default value: 0

### enable\_read\_status\_logic

Enables logic to handle the status register reads as per the '3 Volt Intel StrataFlash Memory' specification.

Type: bool

Default value: false

### layout

Flash layout.

Type: string

Default value: ""



**model\_blocklock**

Model per-block locking and set all the blocks to locked state on reset.

Type: `bool`

Default value: `false`

**size**

Memory Size.

Type: `uint64_t`

Default value: `0x40000`

**trapwrite**

Generate abort on write.

Type: `bool`

Default value: `false`

**unphysical\_writes**

Writes to flash are overwrite not AND.

Type: `bool`

Default value: `true`

## 3.216 InterruptCombiner

Defined in `LISA/InterruptCombiner.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
0	Alpha support

For an explanation of the quality levels, see [Quality level definitions](#).

### About InterruptCombiner

Interrupt Combiner.

### Iris and MTI instances for InterruptCombiner

This model has the following Iris instances:

Name	Instance type
InterruptCombiner	InterruptCombiner

### Ports for InterruptCombiner

Port	Direction	Protocol	Description
apb_bus_s	slave	PVBus	-
aresetn_in	slave	Signal	-
clk_in	slave	ClockSignal	-
irq_in	slave	Signal	-
irq_out	master	Signal	-

### Parameters for InterruptCombiner

#### **diagnostics**

Diagnostics.

Type: `uint8_t`

Default value: 2

#### **no\_clr**

Disable registering and manual clearing of interrupts.

Type: `bool`

Default value: `true`

#### **num\_irqs**

Number of Interrupt to be combined to single interrupt line.

Type: `uint16_t`

Default value: 672

## 3.217 Interrupt\_Router

Defined in `LISA/Interrupt_Router.lisa`.

### About Interrupt\_Router

Interrupt Router Registers.

## Ports for Interrupt\_Router

Port	Direction	Protocol	Description
lockdown	slave	Signal	-
out_interrupts0	master	Signal	-
out_interrupts10	master	Signal	-
out_interrupts11	master	Signal	-
out_interrupts12	master	Signal	-
out_interrupts13	master	Signal	-
out_interrupts14	master	Signal	-
out_interrupts15	master	Signal	-
out_interrupts1	master	Signal	-
out_interrupts2	master	Signal	-
out_interrupts3	master	Signal	-
out_interrupts4	master	Signal	-
out_interrupts5	master	Signal	-
out_interrupts6	master	Signal	-
out_interrupts7	master	Signal	-
out_interrupts8	master	Signal	-
out_interrupts9	master	Signal	-
pvbus_s	slave	PVBus	-
reset_signal	slave	Signal	-
shared_interrupt	slave	Signal	-
tamper_interrupt	master	Signal	-

## Parameters for Interrupt\_Router

### **diagnostics**

Diagnostics.

Type: `uint32_t`

Default value: 0

### **ici\_dst**

Interrupt Controller Destination.

Type: `string`

Default value: N/A

### **ici\_en**

Interrupt Controller Enable.

Type: `string`

Default value: N/A

**lde\_lvl**

Lockdown Extension Level.

Type: uint32\_t

Default value: 2

**num\_ici**

Number of Interrupt Controllers Interrupt interface.

Type: uint32\_t

Default value: 2

**num\_shd\_int**

Number of shared interrupts supported.

Type: uint32\_t

Default value: 2

**ro\_access**

Stream ID of master.

Type: uint32\_t

Default value: 0

**rw\_access**

Stream ID of master with Read Write access.

Type: uint32\_t

Default value: 0

## 3.218 IoTSS3\_ManagerSecurityController

Defined in `LISA/IoTSS3_ManagerSecurityController.lisa`.

### About IoTSS3\_ManagerSecurityController

IoT Subsystem SIE-300 Manager (Master) Security Controller.

### Iris and MTI instances for IoTSS3\_ManagerSecurityController

This model has the following Iris instances:

Name	Instance type
IoTSS3_ManagerSecurityController	IoTSS3_ManagerSecurityController
IoTSS3_ManagerSecurityController.pvbusmodifier	PVBusMapper

This model has the following MTI trace components:

Name	Component type
IoTSS3_ManagerSecurityController.pvbusmodifier	PVBusMapper

### Ports for IoTSS3\_ManagerSecurityController

Port	Direction	Protocol	Description
cfg_nonsec	slave	ValueState	-
cfg_sec_resp	slave	ValueState	-
idau_invalidate_region	slave	Value_64	-
irq	master	StateSignal	-
pvbus_m	master	PVBus	-
pvbus_s	slave	PVBus	-

### Parameters for IoTSS3\_ManagerSecurityController

#### **IRQ\_ENABLE\_RD**

Interrupt enable read.

Type: `bool`

Default value: `true`

#### **IRQ\_ENABLE\_WR**

Interrupt enable write.

Type: `bool`

Default value: `true`

#### **diagnostics**

Diagnostics.

Type: `int32_t`

Default value: `0`

## 3.219 IoTSS3\_MemoryProtectionController

Defined in `LISA/IoTSS3_MemoryProtectionController.lisa`.

### About IoTSS3\_MemoryProtectionController

IoT Subsystem SIE-300 Memory Protection Controller.

### Iris and MTI instances for IoTSS3\_MemoryProtectionController

This model has the following Iris instances:

Name	Instance type
<code>IoTSS3_MemoryProtectionController</code>	<code>IoTSS3_MemoryProtectionController</code>
<code>IoTSS3_MemoryProtectionController.bus_mapper</code>	<code>PVBusMapper</code>
<code>IoTSS3_MemoryProtectionController.busslave</code>	<code>PVBusSlave</code>
<code>IoTSS3_MemoryProtectionController.gating_disabled_thread_event</code>	<code>SchedulerThreadEvent</code>

This model has the following MTI trace components:

Name	Component type
<code>IoTSS3_MemoryProtectionController.bus_mapper</code>	<code>PVBusMapper</code>
<code>IoTSS3_MemoryProtectionController.busslave</code>	<code>PVBusSlave</code>

### Ports for IoTSS3\_MemoryProtectionController

Port	Direction	Protocol	Description
<code>cfg_init_value</code>	slave	<code>ValueState</code>	-
<code>cfg_sec_resp</code>	slave	<code>ValueState</code>	-
<code>config_pvbus_s</code>	slave	<code>PVBus</code>	-
<code>idau_invalidate_region</code>	slave	<code>Value_64</code>	-
<code>mpc_irq</code>	master	<code>StateSignal</code>	-
<code>pvbus_m</code>	master	<code>PVBus</code>	-
<code>pvbus_s</code>	slave	<code>PVBus</code>	-
<code>reset_in</code>	slave	<code>Signal</code>	-

### Parameters for IoTSS3\_MemoryProtectionController

#### **BLK\_MAX**

Maximum block index configuration.

Type: `uint32_t`

Default value: `0xFFFF`

#### **BLK\_SIZE**

Block size configuration.

Type: `uint32_t`

Default value: `0x3`

**GATE\_PRESENT**

Memory gating logic present/not.

Type: `bool`

Default value: `true`

**IRQ\_ENABLE\_RD**

Interrupt enable read.

Type: `bool`

Default value: `true`

**IRQ\_ENABLE\_WR**

Interrupt enable write.

Type: `bool`

Default value: `true`

**diagnostics**

Diagnostics.

Type: `int32_t`

Default value: `0`

3.220 **IoTSS3\_SecureAccessConfig**

Defined in `LISA/IoTSS3_SecureAccessConfig.lisa`.

**About IoTSS3\_SecureAccessConfig**

IoTSS3 Secure Control Register Block.

**Iris and MTI instances for IoTSS3\_SecureAccessConfig**

This model has the following Iris instances:

Name	Instance type
<code>IoTSS3_SecureAccessConfig</code>	<a href="#">IoTSS3_SecureAccessConfig</a>
<code>IoTSS3_SecureAccessConfig.bus_mapper</code>	<a href="#">PVBusMapper</a>

Name	Instance type
IoTSS3_SecureAccessConfig.busslave_ns	PVBusSlave
IoTSS3_SecureAccessConfig.busslave_s	PVBusSlave
IoTSS3_SecureAccessConfig.idau_busmaster	PVBusMaster
IoTSS3_SecureAccessConfig.p_ahb_bus_mapper	PVBusMapper

This model has the following MTI trace components:

Name	Component type
IoTSS3_SecureAccessConfig.bus_mapper	PVBusMapper
IoTSS3_SecureAccessConfig.busslave_ns	PVBusSlave
IoTSS3_SecureAccessConfig.busslave_s	PVBusSlave
IoTSS3_SecureAccessConfig.idau_busmaster	PVBusMaster
IoTSS3_SecureAccessConfig.p_ahb_bus_mapper	PVBusMapper

### Ports for IoTSS3\_SecureAccessConfig

Port	Direction	Protocol	Description
acc_waitn	master	ValueState	-
brg_in	slave	StateSignal	-
brg_out	master	Signal	-
idau	master	PVBus	-
mainnspccexp	master	ValueState	-
mainpppcexp	master	ValueState	-
mem_gating_filter_in	slave	PVBus	-
mem_gating_filter_out	master	PVBus	-
mpc_in	slave	StateSignal	-
mpc_out	master	Signal	-
msc_in	slave	StateSignal	-
msc_out	master	Signal	-
npuspporpl	master	Signal	-
npuspporsl	master	Signal	-
p_ahb_gating_filter_in	slave	PVBus	-
p_ahb_gating_filter_out	master	PVBus	-
periphnsppc0	master	ValueState	-
periphnsppc1	master	ValueState	-
periphnsppcexp	master	ValueState	-
periphpppc0	master	ValueState	-
periphpppc1	master	ValueState	-
periphpppcexp	master	ValueState	-
ppc_in	slave	StateSignal	-
ppc_out	master	Signal	-
pvbuse_nonsecure	slave	PVBus	-



Port	Direction	Protocol	Description
pvbus_secure	slave	PVBus	-
reset_in	slave	Signal	-
security_resp	master	ValueState	-

## Parameters for IoTSS3\_SecureAccessConfig

### **CODENSC**

Whether 0x10000000..0x1FFFFFFF is non-secure-callable.

Type: `bool`

Default value: `false`

### **DISABLE\_GATING**

Disable Memory gating logic.

Type: `bool`

Default value: `false`

### **IGNORE\_MEM\_MAP**

Ignore Memory mapping logic.

Type: `bool`

Default value: `false`

### **MAINPPCEXP\_DIS0**

Disables support for individual bits on the MAINNSPPCEXP0 and MAINPPPCEXP0 buses.

Type: `uint32_t`

Default value: `0x0000`

### **MAINPPCEXP\_DIS1**

Disables support for individual bits on the MAINNSPPCEXP1 and MAINPPPCEXP1 buses.

Type: `uint32_t`

Default value: `0x0000`

### **MAINPPCEXP\_DIS2**

Disables support for individual bits on the MAINNSPPCEXP2 and MAINPPPCEXP2 buses.

Type: `uint32_t`

Default value: 0x0000

### **MAINPPCEXP\_DIS3**

Disables support for individual bits on the MAINNSPPCEXP3 and MAINPPPCEXP3 buses.

Type: uint32\_t

Default value: 0x0000

### **PERIPHPPCEXP\_DIS0**

Disables support for individual bits on the PERIPHNSPPCEXP0 and PERIPHPPPCEXP0 buses.

Type: uint32\_t

Default value: 0x0000

### **PERIPHPPCEXP\_DIS1**

Disables support for individual bits on the PERIPHNSPPCEXP1 and PERIPHPPPCEXP1 buses.

Type: uint32\_t

Default value: 0x0000

### **PERIPHPPCEXP\_DIS2**

Disables support for individual bits on the PERIPHNSPPCEXP2 and PERIPHPPPCEXP2 buses.

Type: uint32\_t

Default value: 0x0000

### **PERIPHPPCEXP\_DIS3**

Disables support for individual bits on the PERIPHNSPPCEXP3 and PERIPHPPPCEXP3 buses.

Type: uint32\_t

Default value: 0x0000

### **RAMNSC**

Whether 0x30000000..0x3FFFFFFF is non-secure-callable.

Type: bool

Default value: false

### **diagnostics**

Diagnostics.

Type: int32\_t

Default value: 0

## 3.221 IoTSS3\_SystemControl

Defined in `LISA/IoTSS3_SystemControl.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
3	Alpha support

For an explanation of the quality levels, see [Quality level definitions](#).

### About IoTSS3\_SystemControl

IoT Subsystem System 3.0 Control registers.

### Iris and MTI instances for IoTSS3\_SystemControl

This model has the following Iris instances:

Name	Instance type
<code>IoTSS3_SystemControl</code>	<code>IoTSS3_SystemControl</code>
<code>IoTSS3_SystemControl.busmaster</code>	<code>PVBusMaster</code>
<code>IoTSS3_SystemControl.busslave</code>	<code>PVBusSlave</code>

This model has the following MTI trace components:

Name	Component type
<code>IoTSS3_SystemControl.busmaster</code>	<code>PVBusMaster</code>
<code>IoTSS3_SystemControl.busslave</code>	<code>PVBusSlave</code>

### Ports for IoTSS3\_SystemControl

Port	Direction	Protocol	Description
<code>busmaster_control</code>	master	<code>PVTransactionMaster</code>	-
<code>cpu0_lockup_reset_request</code>	slave	Signal	-
<code>cpu0_warm_reset_request</code>	slave	Signal	-
<code>cpu0core_ppu_irq</code>	slave	Signal	-
<code>cpu1_lockup_reset_request</code>	slave	Signal	-
<code>cpu1_warm_reset_request</code>	slave	Signal	-
<code>cpu2_lockup_reset_request</code>	slave	Signal	-
<code>cpu2_warm_reset_request</code>	slave	Signal	-
<code>cpu3_lockup_reset_request</code>	slave	Signal	-
<code>cpu3_warm_reset_request</code>	slave	Signal	-
<code>cpuextnmienable_out</code>	master	Signal	-

Port	Direction	Protocol	Description
cpuintnrmienable_out	master	Signal	-
cpuwait_out	master	Signal	-
crypto_ppu_irq	slave	Signal	-
crypto_warm_reset_request	slave	Signal	-
dbg_ppu_irq	slave	Signal	-
dbgen_in	slave	Signal	-
dbgen_out	master	Signal	-
host_level_reset_request	slave	Signal	-
initsvtor	master	Value	-
mgmt_ppu_irq	slave	Signal	-
niden_in	slave	Signal	-
niden_out	master	Signal	-
nonsecure_watchdog_reset_request	slave	Signal	-
np0_ppu_irq	slave	Signal	-
pdc_m_pvb_m	master	PVBus	-
po_reset	master	Signal	-
pvb_s	slave	PVBus	-
secure_watchdog_reset_request	slave	Signal	-
slow_clock_watchdog_reset_request	slave	Signal	-
software_reset_request	slave	Signal	-
spiden_in	slave	Signal	-
spiden_out	master	Signal	-
spniden_in	slave	Signal	-
spniden_out	master	Signal	-
subsystem_hardware_reset_request	slave	Signal	-
sys_ppu_irq	slave	Signal	-
warm_reset	master	Signal	-

## Parameters for IoTSS3\_SystemControl

### INITSVTOR\_RST

Reset int32\_t for INITSVTOR. Should match cpu<i>.INITSVTOR.

Type: uint32\_t

Default value: 0x0

### NUMCPU

Number of Cortex-M CPU cores in the subsystem.

Type: uint8\_t

Default value: 1

**NUMVMBANK**

Number of Volatile Memory Banks.

Type: `uint8_t`

Default value: 2

**SWRESETREQ\_BIT**

CPU 0 Warm Reset Request Enable Default Value.

Type: `uint8_t`

Default value: 9

**cpu0wait**

Whether to hold cpu0 in reset at boot.

Type: `bool`

Default value: `false`

**cpu1wait**

Whether to hold cpu1 in reset at boot.

Type: `bool`

Default value: `true`

**cpu2wait**

Whether to hold cpu2 in reset at boot.

Type: `bool`

Default value: `true`

**cpu3wait**

Whether to hold cpu3 in reset at boot.

Type: `bool`

Default value: `true`

**diagnostics**

Type: `uint8_t`

Default value: 2

## 3.222 IoTSS\_AccessControlGate

Defined in `LISA/IoTSS_AccessControlGate.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
1	Alpha support

For an explanation of the quality levels, see [Quality level definitions](#).

### About IoTSS\_AccessControlGate

IoT Subsystem Access Control Gate.

### Iris and MTI instances for IoTSS\_AccessControlGate

This model has the following Iris instances:

Name	Instance type
<code>IoTSS_AccessControlGate</code>	<a href="#">IoTSS_AccessControlGate</a>
<code>IoTSS_AccessControlGate.bus_mapper</code>	<a href="#">PVBusMapper</a>

This model has the following MTI trace components:

Name	Component type
<code>IoTSS_AccessControlGate.bus_mapper</code>	<a href="#">PVBusMapper</a>

### Ports for IoTSS\_AccessControlGate

Port	Direction	Protocol	Description
<code>ext_gate</code>	slave	<a href="#">Signal</a>	-
<code>master_ppuhwstat</code>	slave	<a href="#">Value</a>	-
<code>pvbus_m</code>	master	<a href="#">PVBus</a>	-
<code>pvbus_s</code>	slave	<a href="#">PVBus</a>	-
<code>slave_ppuhwstat</code>	slave	<a href="#">Value</a>	-
<code>wake_request</code>	master	<a href="#">Signal</a>	-

### Parameters for IoTSS\_AccessControlGate

#### **enabled**

Enable the ACG. If disabled, will let all transactions through without side effects.

Type: `bool`

Default value: `true`

## 3.223 IoTSS\_PeripheralProtectionController

Defined in `LISA/IoTSS_PeripheralProtectionController.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
1	Alpha support

For an explanation of the quality levels, see [Quality level definitions](#).

### About IoTSS\_PeripheralProtectionController

IoT Subsystem Peripheral Protection Controller.

### Iris and MTI instances for IoTSS\_PeripheralProtectionController

This model has the following Iris instances:

Name	Instance type
<code>IoTSS_PeripheralProtectionController</code>	<a href="#">IoTSS_PeripheralProtectionController</a>
<code>IoTSS_PeripheralProtectionController.bus_mapperX</code> (where X = 0–15)	<a href="#">PVBusMapper</a>

This model has the following MTI trace components:

Name	Component type
<code>IoTSS_PeripheralProtectionController.bus_mapperX</code> (where X = 0–15)	<a href="#">PVBusMapper</a>

### Ports for IoTSS\_PeripheralProtectionController

Port	Direction	Protocol	Description
<code>cfg_ap</code>	slave	<a href="#">ValueState</a>	-
<code>cfg_nonsec</code>	slave	<a href="#">ValueState</a>	-
<code>cfg_sec_resp</code>	slave	<a href="#">ValueState</a>	-
<code>idau_invalidate_region</code>	slave	<a href="#">Value_64</a>	-
<code>ppc_irq</code>	master	<a href="#">StateSignal</a>	-
<code>pvbus_m</code>	master	<a href="#">PVBus</a>	-
<code>pvbus_s</code>	slave	<a href="#">PVBus</a>	-

### Parameters for IoTSS\_PeripheralProtectionController

#### **DISABLE\_GATING**

Disable Memory gating logic.

Type: `bool`

Default value: `false`

**NONSEC\_MASK**

16-bit wide mask for security checking of ports: 0 = check, 1 = mask.

Type: uint32\_t

Default value: 0x0000

**PORTx\_ENABLE**

Enable (1) or disable (0) port x (where x is between 0-15): enable = 1, disable = 0.

Type: uint32\_t

Default value: 0xFFFF

**diagnostics**

Diagnostics.

Type: int32\_t

Default value: 0

## 3.224 Juno\_sysregs

Defined in LISA/Juno\_sysregs.lisa.

**About Juno\_sysregs**

IOFPGA system register unit.

**Iris and MTI instances for Juno\_sysregs**

This model has the following Iris instances:

Name	Instance type
Juno_sysregs	Juno_sysregs
Juno_sysregs.pvbuslave	PVBusSlave

This model has the following MTI trace components:

Name	Component type
Juno_sysregs.pvbuslave	PVBusSlave

**Ports for Juno\_sysregs**

Port	Direction	Protocol	Description
clock_100Hz	slave	ClockSignal	-
clock_24Mhz	slave	ClockSignal	-



Port	Direction	Protocol	Description
ethernet_irq	master	Signal	-
mmc_presence	slave	StateSignal	-
pb_irq	master	Signal	-
pvbus	slave	PVBus	-
rtcc_irq	master	Signal	-
tile1_irq	master	Signal	-
usb_irq	master	Signal	-

## Parameters for Juno\_sysregs

### **diagnostics**

Diagnostics.

Type: uint32\_t

Default value: 0

### **pcie\_mac**

PCIe MAC address.

Type: uint64\_t

Default value: 0x0002f7000001

### **rev**

Board revision.

Type: uint32\_t

Default value: 0

## 3.225 KeyManagementUnit

Defined in `LISA/KeyManagementUnit.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
0.5	Alpha support

For an explanation of the quality levels, see [Quality level definitions](#).

### About KeyManagementUnit

KeyManagementUnit.

## Iris and MTI instances for KeyManagementUnit

This model has the following Iris instances:

Name	Instance type
KeyManagementUnit	KeyManagementUnit
KeyManagementUnit.apb	PVBusSlave
KeyManagementUnit.keys_in	PVBusSlave
KeyManagementUnit.keys_out[0].pvbusmaster	PVBusMaster

This model has the following MTI trace components:

Name	Component type
KeyManagementUnit.apb	PVBusSlave
KeyManagementUnit.keys_in	PVBusSlave
KeyManagementUnit.keys_out[0].pvbusmaster	PVBusMaster

## Ports for KeyManagementUnit

Port	Direction	Protocol	Description
apb	slave	PVBus	register access via apb port
hw_keys_in	slave	PVBus	HW key register access via private apb port - connect to LCM
irq_out	master	Signal	IRQ signal out
keys_out	master	PVBus	output keys via port
reset_in	slave	Signal	Reset signal in

## Parameters for KeyManagementUnit

### KMUDKPARV0

Reset value of the KMUDKPA<0> register. If no hardware key slot is supported (KMUNHWKSLTS is 0), this configuration option must be set to 0x0000\_0000.

Type: uint32\_t

Default value: 0x0

### KMUDKPARV1

Reset value of the KMUDKPA<1> register. If no hardware key slot is supported (KMUNHWKSLTS is 1), this configuration option must be set to 0x0000\_0000.

Type: uint32\_t

Default value: 0x0

### KMUDKPARV2

Reset value of the KMUDKPA<2> register. If no hardware key slot is supported (KMUNHWKSLTS is 2), this configuration option must be set to 0x0000\_0000.

Type: `uint32_t`

Default value: `0x0`

### **KMUDKPARV3**

Reset value of the KMUDKPA<3> register. If no hardware key slot is supported (KMUNHWKSLTS is 3), this configuration option must be set to `0x0000_0000`.

Type: `uint32_t`

Default value: `0x0`

### **KMUDKPARV4**

Reset value of the KMUDKPA<4> register. If no hardware key slot is supported (KMUNHWKSLTS is 4), this configuration option must be set to `0x0000_0000`.

Type: `uint32_t`

Default value: `0x0`

### **KMUDKPARV5**

Reset value of the KMUDKPA<5> register. If no hardware key slot is supported (KMUNHWKSLTS is 5), this configuration option must be set to `0x0000_0000`.

Type: `uint32_t`

Default value: `0x0`

### **KMUDKPARV6**

Reset value of the KMUDKPA<6> register. If no hardware key slot is supported (KMUNHWKSLTS is 6), this configuration option must be set to `0x0000_0000`.

Type: `uint32_t`

Default value: `0x0`

### **KMUDKPARV7**

Reset value of the KMUDKPA<7> register. If no hardware key slot is supported (KMUNHWKSLTS is 7), this configuration option must be set to `0x0000_0000`.

Type: `uint32_t`

Default value: `0x0`

### **KMUKSCRVO**

Reset value of the KMUKSC<0> register. If no hardware key slot is supported (KMUNHWKSLTS is 0), this configuration option must be set to `0x0000_0000`.

Type: `uint32_t`

Default value: `0x0`

#### **KMUKSCRV1**

Reset value of the KMUKSC<1> register. If no hardware key slot is supported (KMUNHWKSLTS is 1), this configuration option must be set to `0x0000_0000`.

Type: `uint32_t`

Default value: `0x0`

#### **KMUKSCRV2**

Reset value of the KMUKSC<2> register. If no hardware key slot is supported (KMUNHWKSLTS is 2), this configuration option must be set to `0x0000_0000`.

Type: `uint32_t`

Default value: `0x0`

#### **KMUKSCRV3**

Reset value of the KMUKSC<3> register. If no hardware key slot is supported (KMUNHWKSLTS is 3), this configuration option must be set to `0x0000_0000`.

Type: `uint32_t`

Default value: `0x0`

#### **KMUKSCRV4**

Reset value of the KMUKSC<4> register. If no hardware key slot is supported (KMUNHWKSLTS is 4), this configuration option must be set to `0x0000_0000`.

Type: `uint32_t`

Default value: `0x0`

#### **KMUKSCRV5**

Reset value of the KMUKSC<5> register. If no hardware key slot is supported (KMUNHWKSLTS is 5), this configuration option must be set to `0x0000_0000`.

Type: `uint32_t`

Default value: `0x0`

#### **KMUKSCRV6**

Reset value of the KMUKSC<6> register. If no hardware key slot is supported (KMUNHWKSLTS is 6), this configuration option must be set to `0x0000_0000`.

Type: `uint32_t`

Default value: `0x0`

#### **KMUKSCR7**

Reset value of the KMUKSC<7> register. If no hardware key slot is supported (KMUNHWKSLTS is 7), this configuration option must be set to `0x0000_0000`.

Type: `uint32_t`

Default value: `0x0`

#### **KMUNHWKSLTS**

KMU Number of hardware key slots (0..8).

Type: `uint32_t`

Default value: 7

#### **KMUNKS**

KMU number of key slots (0x1=2, 0x2=4, 0x3=8, 0x4=16, 0x5=32).

Type: `uint32_t`

Default value: `0x5`

#### **KMUTANG**

KMU Address range for secure devices. See docs for info.

Type: `uint32_t`

Default value: `0x5`

#### **diagnostics**

diagnostics.

Type: `int32_t`

Default value: 2

## 3.226 Kits2\_Timer

Defined in `LISA/Kits2_Timer.lisa`.

### About Kits2\_Timer

Kits2 Timer.

## Iris and MTI instances for Kits2\_Timer

This model has the following Iris instances:

Name	Instance type
Kits2_Timer	Kits2_Timer
Kits2_Timer.clk_div	ClockDivider
Kits2_Timer.counter	CounterModule
Kits2_Timer.counter.bussubordinate	PVBusSlave

This model has the following MTI trace components:

Name	Component type
Kits2_Timer.clk_div	ClockDivider
Kits2_Timer.counter.bussubordinate	PVBusSlave

## Ports for Kits2\_Timer

Port	Direction	Protocol	Description
clock	slave	ClockSignal	-
enable	slave	Signal	-
irq_out	master	Signal	-
timer_freq	slave	ClockRateControl	-
timer_value	slave	ValueState	-

## Parameters for Kits2\_Timer

### **clk\_div.div**

Clock Rate Divider.

Type: uint64\_t

Default value: 1

### **clk\_div.mul**

Clock Rate Multiplier.

Type: uint64\_t

Default value: 1

### **diagnostics**

Diagnostics.

Type: uint32\_t

Default value: 0

**timer\_interval**

Timer ticks to count before firing interrupt.

Type: uint32\_t

Default value: 0

## 3.227 LCD2SystemC

Defined in `examples/SystemCExport/Bridges/LCD2SystemC.lisa`.

### About LCD2SystemC

Converts LCD protocol to SystemC.

### Iris and MTI instances for LCD2SystemC

This model has the following Iris instances:

Name	Instance type
LCD2SystemC	LCD2SystemC

### Ports for LCD2SystemC

Port	Direction	Protocol	Description
all_received_spl	master	AMBAPVSignal	-
all_received_u	master	AMBAPVSignal	-
lcd_s	slave	LCD	-
lock_m	master	AMBAPVValueState64	-
setPreferredLayout_d	master	AMBAPVValue	-
setPreferredLayout_h	master	AMBAPVValue	-
setPreferredLayout_w	master	AMBAPVValue	-
unlock_m	master	AMBAPVSignal	-
update_h	master	AMBAPVValue	-
update_w	master	AMBAPVValue	-
update_x	master	AMBAPVValue	-
update_y	master	AMBAPVValue	-

### Parameters for LCD2SystemC

This component does not have any parameters.

## 3.228 LS64TestingFIFO

Defined in `examples/LISA/Common/LISA/LS64TestingFIFO.lisa`.

### About LD64TestingFIFO

LS64TestingFIFO is a LISA component for testing the FEAT\_LS64 architectural feature. It accepts ST64B instructions and places the supplied data into a configurable buffer. LD64B instructions can then read this data out of the FIFO. The value returned can configurably be bitwise inverted.

It also supports the ST64BV variants where success and failure are reported by a return result rather than by transaction success and failure.

### Iris and MTI instances for LS64TestingFIFO

This model has the following Iris instances:

Name	Instance type
LS64TestingFIFO	LS64TestingFIFO
LS64TestingFIFO.pvbuslave	PVBusSlave

This model has the following MTI trace components:

Name	Component type
LS64TestingFIFO.pvbuslave	PVBusSlave

### Ports for LS64TestingFIFO

Port	Direction	Protocol	Description
pvbus_s	slave	PVBus	Bus subordinate interface.

### Parameters for LS64TestingFIFO

#### **buffer\_size**

The number of 64-byte slots in the FIFO.

Type: unsigned

Default value: 2

#### **op\_type**

The operation performed on the 64-byte transaction data 0 - None, 1 - Bitwise Negate.

Type: int

Default value: 1



## 3.229 Labeller

Defined in `LISA/Labeller.lisa`.

### About Labeller

Labeller and LabellerForDMA330 are utility components that allow the system designer to embed values into the Label field for transactions generated by a Bus Master. They are located between PVBUS Master and Slave ports.

As FastModels have no direct concept of AXI ID, those components that use AXI ID information have to use a proxy for it.

By default, Labeller utilizes bits [31:16] of the component's `ManagerID` to store a label (see `PVMemoryAttributes.h`). Additionally, the `use_msb_for_manager_id` parameter is available, enabling the use of bits [63:48] of the `ManagerID` to differentiate transactions initiated by distinct managers.

Those components that need to know an analog of AXI ID should have a configurable mapping from 'label' to its internal representation of AXI ID.

When assembling a SoC the designer has to place a labeller under every component that has to be distinguished and assign it a unique label.

The following example creates a labeller to add an ID for an HDLCD controller that is upstream of a TZC\_400. The system designer specifies a unique set of IDs for use as Non-Secure Access IDs (NSAIDs) in the TZC\_400. The labeller can insert these IDs directly into the transaction.

```
p1370_hdlcd : PL370_HDLCDC();
hdlcd_labeller : Labeller( "label" = 2 );
p1370_hdlcd.pvbus_m => hdlcd_labeller.pvbus_s;
hdlcd_labeller.pvbus_m => output_bus.pvbus_s;
```

### Iris and MTI instances for Labeller

This model has the following Iris instances:

Name	Instance type
Labeller	Labeller
Labeller.pvbusmodifier	PVBusMapper

This model has the following MTI trace components:

Name	Component type
Labeller.pvbusmodifier	PVBusMapper

### Ports for Labeller

Port	Direction	Protocol	Description
pvbus_m	master	PVBus	Output with modified ManagerID.
pvbus_s	slave	PVBus	Unmodified input.

## Parameters for Labeller

### **diagnostics**

Diagnostics 0=Disable, 1=Warnings, 2= Information (i.e., useful messages), 3=Debug information (i.e., all traces).

Type: `uint32_t`

Default value: 0

### **label**

The label to apply to all transactions flowing through the labeller.

Type: `uint16_t`

Default value: 0

### **use\_msb\_for\_manager\_id**

Use bits 63:48 for unique IDs.

Type: `bool`

Default value: `false`

## 3.230 LabellerForDMA330

Defined in `LISA/LabellerForDMA330.lisa`.

### About LabellerForDMA330

As FastModels have no direct concept of AXI ID those components that use AXI ID information have to use a proxy for it.

We use the [31:16] bits of the component's `ManagerID` to store a label (see `PVMemoryAttributes.h`)

Those components that need to know an analog of AXI ID should have a configurable mapping from `label` to its internal representation of AXI ID.

When assembling a SoC the designer has to place a Labeller under every component that has to be distinguished and assign it a unique label.

This specific labeller understands the `ManagerID` used by the instruction stream and uses a different label for it. It also provides the option of discriminating the DMA-330 data channels.

### Iris and MTI instances for LabellerForDMA330

This model has the following Iris instances:

Name	Instance type
LabellerForDMA330	LabellerForDMA330
LabellerForDMA330.pvbusmapper	PVBusMapper

This model has the following MTI trace components:

Name	Component type
LabellerForDMA330.pvbusmapper	PVBusMapper

### Ports for LabellerForDMA330

Port	Direction	Protocol	Description
pvbus_m	master	PVBus	Output with modified ManagerID.
pvbus_s	slave	PVBus	Unmodified input.

### Parameters for LabellerForDMA330

#### **dma330\_data\_label**

The label to apply to all *data* transactions flowing through the labeller. Used as a base value in conjunction with the channel ID if data-channel discrimination is enabled.

Type: `uint16_t`

Default value: `0`

#### **dma330\_discriminate\_data\_channels**

Discriminate between DMA-330 data channels. Channel ID is added to the data label.

Type: `bool`

Default value: `false`

#### **dma330\_ns\_instruction\_label**

The label to apply to all non-secure *instructions* transactions flowing through the labeller.

Type: `uint16_t`

Default value: `0`

#### **dma330\_s\_instruction\_label**

The label to apply to all secure *instructions* transactions flowing through the labeller.

Type: `uint16_t`

Default value: `0`

## 3.231 LabellerForGPUProtMode

Defined in `LISA/LabellerForGPUProtMode.lisa`.

### About LabellerForGPUProtMode

This component adds Non-Secure Access IDs (NSAIDs) to the transactions generated by the GPU. The NSAID is a four-bit number. It allows other components, such as a TrustZone Controller (TZC) or a Dynamic Memory Controller (DMC) to filter transactions and control access to memory regions that are designated as protected.

Certain Bifrost GPUs support a protected mode of operation intended to stop valuable or 'protected' data, for example the decoded frames of a DRM protected movie being written to memory that is generally accessible.

They tell the rest of the system they are in this mode by setting the signal `PROTMODE`.

External hardware outside the GPU must respond to this by making whatever adjustment is required to ensure the content goes to memory that is not generally accessible.

This labeller represents such hardware in an effort to ensure it is not forgotten in the corresponding RTL.

### Iris and MTI instances for LabellerForGPUProtMode

This model has the following Iris instances:

Name	Instance type
<code>LabellerForGPUProtMode</code>	<a href="#">LabellerForGPUProtMode</a>
<code>LabellerForGPUProtMode.pvbusmodifier</code>	<a href="#">PVBusMapper</a>

This model has the following MTI trace components:

Name	Component type
<code>LabellerForGPUProtMode.pvbusmodifier</code>	<a href="#">PVBusMapper</a>

### Ports for LabellerForGPUProtMode

Port	Direction	Protocol	Description
<code>prot_mode</code>	slave	<a href="#">Signal</a>	Input to determine whether output is supposed to be protected or not
<code>pvbus_m</code>	master	<a href="#">PVBus</a>	Output with modified ManagerID.
<code>pvbus_s</code>	slave	<a href="#">PVBus</a>	Unmodified input.

### Parameters for LabellerForGPUProtMode

#### **gpu\_id\_normal**

NSAID to apply to all transactions flowing through the labeller when `prot_mode` is low.

Type: `uint32_t`

Default value: 0

### **gpu\_id\_protected**

NSAID to apply to all transactions flowing through the labeller when prot\_mode is high.

Type: uint32\_t

Default value: 0

## 3.232 LabellerIdauSecurity

Defined in LISA/LabellerIdauSecurity.lisa.

### **Iris and MTI instances for LabellerIdauSecurity**

This model has the following Iris instances:

Name	Instance type
LabellerIdauSecurity	LabellerIdauSecurity
LabellerIdauSecurity.idau_busmaster	PVBusMaster
LabellerIdauSecurity.pvbusmodifier	PVBusMapper
LabellerIdauSecurity.remap_busmaster	PVBusMaster

This model has the following MTI trace components:

Name	Component type
LabellerIdauSecurity.idau_busmaster	PVBusMaster
LabellerIdauSecurity.pvbusmodifier	PVBusMapper
LabellerIdauSecurity.remap_busmaster	PVBusMaster

### **Ports for LabellerIdauSecurity**

Port	Direction	Protocol	Description
idau_invalidate_region	slave	Value_64	-
idau	master	PVBus	-
pvbus_m	master	PVBus	-
pvbus_s	slave	PVBus	-

### **Parameters for LabellerIdauSecurity**

#### **diagnostics**

Diagnostics.

Type: int32\_t

Default value: 0

## 3.233 LabellerManagerIdExtendedIdUserFlag

Defined in `LISA/LabellerManagerIdExtendedIdUserFlags.lisa`.

### About LabellerManagerIdExtendedIdUserFlag

LabellerManagerIdExtendedIdUserFlag is a utility component that allows the system designer to modify the ManagerID, ExtendedID, and UserFlags attributes of PVBUS transactions.

### Iris and MTI instances for LabellerManagerIdExtendedIdUserFlag

This model has the following Iris instances:

Name	Instance type
LabellerManagerIdExtendedIdUserFlag	LabellerManagerIdExtendedIdUserFlag
LabellerManagerIdExtendedIdUserFlag.pvbuslogger	PVBusLogger
LabellerManagerIdExtendedIdUserFlag.pvbuslogger.mapper	PVBusMapper
LabellerManagerIdExtendedIdUserFlag.pvbusmodifier	PVBusMapper

This model has the following MTI trace components:

Name	Component type
LabellerManagerIdExtendedIdUserFlag.pvbuslogger	PVBusLogger
LabellerManagerIdExtendedIdUserFlag.pvbuslogger.mapper	PVBusMapper
LabellerManagerIdExtendedIdUserFlag.pvbusmodifier	PVBusMapper

### Ports for LabellerManagerIdExtendedIdUserFlag

Port	Direction	Protocol	Description
pvbus_m	master	PVBus	Output with modified properties.
pvbus_s	slave	PVBus	Unmodified input.

### Parameters for LabellerManagerIdExtendedIdUserFlag

#### **extendedid**

ExtendedID value to be applied to transactions.

Type: `uint64_t`

Default value: 0

#### **extendedid\_mask**

Mask used to determine which bits of extendedid parameter to be set in the transactions ExtendedID attribute. `0xFFFFFFFFFFFFFFFF` will overwrite all the incoming ExtendedID bits with the value of the extendedid parameter, `0x0` will overwrite none.

Type: `uint64_t`

Default value: 0

**managerid**

ManagerID value to be applied to transactions.

Type: uint64\_t

Default value: 0

**managerid\_mask**

Mask used to determine which bits of managerid parameter to be set in the transactions ManagerID attribute. 0xFFFFFFFFFFFFFFFF will overwrite all the incoming ManagerID bits with the value of the managerid parameter, 0x0 will overwrite none.

Type: uint32\_t

Default value: 0

**userflags**

UserFlags value to be applied to transactions.

Type: uint32\_t

Default value: 0

**userflags\_mask**

Mask used to determine which bits of userflags parameter to be set in the transactions UserFlags attribute. 0xFFFFFFFF will overwrite all the incoming UserFlags bits with the value of the userflags parameter, 0x0 will overwrite none.

Type: uint32\_t

Default value: 0

3.234 LabellerPAS

Defined in LISA/LabellerPAS.lisa.

Iris and MTI instances for LabellerPAS

This model has the following Iris instances:

Name	Instance type
LabellerPAS	LabellerPAS

## Ports for LabellerPAS

Port	Direction	Protocol	Description
pdbus_m	master	PVBus	Output txn PAS modified bus
pdbus_s	slave	PVBus	Input bus

## Parameters for LabellerPAS

### diagnostics

Diagnostics. 0: FatalError, 1:Error, 2:Warning, 3:Info, 4:Debug.

Type: uint8\_t

Default value: 1

### pas

Physical Address Space. 0:Secure, 1:Non Secure, 2:Root, 3:Realm, 4:Non-Secure Protected, 5:System Agent, 6:NA6, 7:NA7.

Type: uint8\_t

Default value: 0x1

## 3.235 LabellerUserSignals

Defined in LISA/LabellerUserSignals.lisa.

### Iris and MTI instances for LabellerUserSignals

This model has the following Iris instances:

Name	Instance type
LabellerUserSignals	LabellerUserSignals
LabellerUserSignals.pdbuslogger	PVBusLogger
LabellerUserSignals.pdbuslogger.mapper	PVBusMapper
LabellerUserSignals.pdbusmodifier	PVBusMapper

This model has the following MTI trace components:

Name	Component type
LabellerUserSignals.pdbuslogger	PVBusLogger
LabellerUserSignals.pdbuslogger.mapper	PVBusMapper
LabellerUserSignals.pdbusmodifier	PVBusMapper



## Ports for LabellerUserSignals

Port	Direction	Protocol	Description
pvbuser_m	master	PVBus	Output with modified UserFlags.
pvbuser_s	slave	PVBus	Unmodified input.

## Parameters for LabellerUserSignals

### user

User signal to be applied to transactions.

Type: int

Default value: 0

## 3.236 LifeCycleManager

Defined in LISA/LifeCycleManager.lisa.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
1.02	Alpha support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following parameters were added:

- rse\_1\_6

### About LifeCycleManager

LifeCycleManager.

### Iris and MTI instances for LifeCycleManager

This model has the following Iris instances:

Name	Instance type
LifeCycleManager	RSE_LifeCycleManager
LifeCycleManager.apb	PVBusSlave
LifeCycleManager.hw_keys_out	PVBusMaster
LifeCycleManager.nvm_external_out	PVBusMaster

This model has the following MTI trace components:

Name	Component type
LifeCycleManager.apb	PVBusSlave
LifeCycleManager.hw_keys_out	PVBusMaster
LifeCycleManager.nvm_external_out	PVBusMaster

### Ports for LifeCycleManager

Port	Direction	Protocol	Description
apb	slave	PVBus	APB4 Subordinate Interface - Access to LCM registers and internal OTP/NVM
dcu_force_disable_in	slave	Value	LCM Force disable DCU_EN signals
hw_keys_out	master	PVBus	APB manager interface for output of the HW Root of Trust keys
lcm_diagnostic_mode_trig_ack_in	slave	Signal	LCM diagnostic mode trigger response from DMA.
lcm_diagnostic_mode_trig_req_out	master	Signal	LCM diagnostic mode trigger request to DMA.
lcm_mission_mode_trig_ack_in	slave	Signal	LCM mission mode trigger response from DMA.
lcm_mission_mode_trig_req_out	master	Signal	LCM mission mode trigger request to DMA.
lcm_mission_provisioning_mode_trig_ack_in	slave	Signal	LCM mission-provisioning mode trigger response from DMA.
lcm_mission_provisioning_mode_trig_req_out	master	Signal	LCM mission-provisioning mode trigger request to DMA.
lcm_mission_se_mode_trig_ack_in	slave	Signal	LCM security-enabled life cycle mode trigger response from DMA.
lcm_mission_se_mode_trig_req_out	master	Signal	LCM security-enabled life cycle mode trigger request to DMA.
lcm_seed_lfsr_data_in	slave	Value_64	-
lcm_seed_lfsr_valid_in	slave	Signal	-
lcs_valid_out	master	Signal	LCM state valid out
lcs	master	Value	The 3-bit value of the LCS register
nvm_external_out	master	PVBus	APB manager interface for external NVM (OTP)
psi_dcu_en0_out	master	Signal	-
psi_dcu_en1_out	master	Signal	Debug Control Enable Values (LCM DCUs). Force disabled Debug Control Enable Values. These signals are driven by the computed fd_dcu_en[5:0] signals..
psi_dcu_gppc_out	master	Signal	General-Purpose Persistent Configuration. These signals are driven by the LCM gppc[15:0] signals and can be exposed to the SoC.
reset_in	slave	Signal	LCM reset in
sp_reset_out	master	Signal	LCM SP_RST_REQ signal out

### Parameters for LifeCycleManager

#### DCU\_PERMANENT\_DISABLE\_MASK\_VAL

Permanently disables the DCU\_EN ports.

Type: `string`

Default value: `"0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF"`

#### **DCU\_SP\_DISABLE\_MASK\_VAL**

The Secure Provisioning disable mask of the 128-bit DCU signals. Clearing a bit in the mask will force the relevant DCU signal to zero when LCM is in Secure Provisioning Mode (SP\_EN=1).

Type: `string`

Default value: `"0x000000000000000000000000AAAAaaaa"`

#### **DISABLE\_DIRECT\_KEY\_APB\_MASKING**

When set to 1, the Direct Key APB masking feature should be disabled.

Type: `uint32_t`

Default value: 0

#### **KRTL\_VAL**

The Krtl value.

Type: `string`

Default value: N/A

#### **OTP\_ADDR\_WIDTH**

The OTP bus width (width in bits = OTP\_ADDR\_WIDTH + 2).

Type: `uint32_t`

Default value: 12

#### **OTP\_MASK\_VAL**

This value must be generated using a true random number generator (1536-bit hex).

Type: `string`

Default value: N/A

#### **OTP\_SIZE\_IN\_BYTES**

The size of the OTP region accessible from the LCM APB-S interface. The maximum size is 60KB.

Type: `uint32_t`

Default value: 16384

**OVERRIDE\_LCM\_DCU\_FORCE\_DISABLE**

If this parameter is non-zero, the LCM\_DCU\_FORCE\_DISABLE register is effectively overwritten with the supplied value. When it is zero, there is no override and the LCM\_DCU\_FORCE\_DISABLE register value is used.

Type: uint32\_t

Default value: 0xAAAAaaaa

 **PCI\_DCU\_PERMANENT\_DISABLE\_MASK\_VAL\_LCS\_CM**

Defines the permanent disable mask for Debug Control Unit (DCU) in Chip Manufacture (CM) Lifecycle State (LCS). (128-bit hex).

Type: string

Default value: "0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF"

 **PCI\_DCU\_PERMANENT\_DISABLE\_MASK\_VAL\_LCS\_DM**

Defines the permanent disable mask for Debug Control Unit (DCU) in Device Manufacture (DM) Lifecycle State (LCS). (128-bit hex).

Type: string

Default value: "0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFeFFFFFFf"

 **PCI\_DCU\_PERMANENT\_DISABLE\_MASK\_VAL\_LCS\_RMA**

Defines the permanent disable mask for Debug Control Unit (DCU) in Return Merchandise Authorization (RMA) Lifecycle State (LCS). (128-bit hex).

Type: string

Default value: "0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFeFFFFFFf"

 **PCI\_DCU\_PERMANENT\_DISABLE\_MASK\_VAL\_LCS\_SECURE**

Defines the permanent disable mask for Debug Control Unit (DCU) in Secure Lifecycle State (LCS). (128-bit hex).

Type: string

Default value: "0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFeFFFFFFf"

 **PCI\_DEFAULT\_DCU\_VAL\_LCS\_CM**

Defines the default Debug Control Unit (DCU) values to load in Chip Manufacture (CM) Lifecycle State (LCS). By default, all debugging is enabled in CM. (128-bit hex).

Type: string

Default value: "0xBFFFFFFFFFFFFFFFFFFFFFFFFF55555555"

**PCI\_DEFAULT\_DCU\_VAL\_LCS\_DM**

Defines the default Debug Control Unit (DCU) values to load in Device Manufacture (DM) Lifecycle State (LCS). By default, all debugging is enabled in DM. (128-bit hex).

Type: string

Default value: "0x80000000000000000000000005655555"

**PCI\_DEFAULT\_DCU\_VAL\_LCS\_RMA**

Defines the default Debug Control Unit (DCU) values to load in Return Merchandise Authorization (RMA) Lifecycle State (LCS). By default, all debugging is enabled in RMA. (128-bit hex).

Type: string

Default value: "0xBFFFFFFF55555555"

**PCI\_DEFAULT\_DCU\_VAL\_LCS\_SECURE**

Defines the default Debug Control Unit (DCU) values to load in Secure Lifecycle State (LCS). By default, all debugging is disabled in SE. (128-bit hex).

Type: string

Default value: "0x000000000000000000000000AAAAaaaa"

**TCI\_DCU\_PERMANENT\_DISABLE\_MASK\_VAL\_LCS\_CM**

Defines the permanent disable mask for Debug Control Unit (DCU) in Chip Manufacture (CM) Lifecycle State (LCS). (128-bit hex).

Type: string

Default value: "0xFFFFFFFFFFFFFFFFFFFFFFFF"

**TCI\_DCU\_PERMANENT\_DISABLE\_MASK\_VAL\_LCS\_DM**

Defines the permanent disable mask for Debug Control Unit (DCU) in Device Manufacture (DM) Lifecycle State (LCS). (128-bit hex).

Type: string

Default value: "0xFFFFFFFFFFFFFFFFFFFFFFFF"

**TCI\_DCU\_PERMANENT\_DISABLE\_MASK\_VAL\_LCS\_RMA**

Defines the permanent disable mask for Debug Control Unit (DCU) in Return Merchandise Authorization (RMA) Lifecycle State (LCS). (128-bit hex).

Type: string

Default value: "0xFFFFFFFFFFFFFFFFFFFFFFFF"

**TCI\_DCU\_PERMANENT\_DISABLE\_MASK\_VAL\_LCS\_SECURE**

Defines the permanent disable mask for Debug Control Unit (DCU) in Secure Lifecycle State (LCS). (128-bit hex).

Type: string

Default value: "0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF"

**TCI\_DEFAULT\_DCU\_VAL\_LCS\_CM**

Defines the default Debug Control Unit (DCU) values to load in Chip Manufacture (CM) Lifecycle State (LCS). By default, all debugging is enabled in CM. (128-bit hex).

Type: string

Default value: "0xBFFFFFFFFFFFFFFFFFFFFFFFFF55555555"

**TCI\_DEFAULT\_DCU\_VAL\_LCS\_DM**

Defines the default Debug Control Unit (DCU) values to load in Device Manufacture (DM) Lifecycle State (LCS). By default, all debugging is enabled in DM. (128-bit hex).

Type: string

Default value: "0xBFFFFFFFFFFFFFFFFFFFFFFFFF55555555"

**TCI\_DEFAULT\_DCU\_VAL\_LCS\_RMA**

Defines the default Debug Control Unit (DCU) values to load in Return Merchandise Authorization (RMA) Lifecycle State (LCS). By default, all debugging is enabled in RMA. (128-bit hex).

Type: string

Default value: "0xBFFFFFFFFFFFFFFFFFFFFFFFFF55555555"

**TCI\_DEFAULT\_DCU\_VAL\_LCS\_SECURE**

Defines the default Debug Control Unit (DCU) values to load in Secure Lifecycle State (LCS). By default, all debugging is disabled in SE. (128-bit hex).

Type: string

Default value: "0xBFFFFFFFFFFFFFFFFFFFFFFFFF55555555"

**VIRGIN\_DCU\_PERMANENT\_DISABLE\_MASK\_VAL\_LCS\_CM**

Defines the disable mask for Debug Control Unit (DCU) values to load in Virgin Chip Manufacture (CM) Lifecycle State (LCS).

Type: string

Default value: "0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF"

**VIRGIN\_DEFAULT\_DCU\_VAL\_LCS\_CM**

Defines the default Debug Control Unit (DCU) values to load in Virgin Chip Manufacture (CM) Lifecycle State (LCS).

Type: `string`

Default value: `"0xBFFFFFFF55555555"`

**diagnostics**

Diagnostics.

Type: `uint32_t`

Default value: `0`

**rse\_1\_6**

`rse_1_6`.

Type: `bool`

Default value: `false`

## 3.237 MHU320AE

Defined in `LISA/MHU320AE.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
MHU320AE	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

**Limitations**

The following features are not yet implemented:

Architectural extensions \* TrustZone Extension (TZE) \* Realm Management Extension (RME) \* Reliability, Availability and Serviceability Extensions (RASE)

Functional safety support \* Lock-step of core MHU logic blocks \* RAM protection \* AMBA AXI5-Stream or ACE5-Lite interconnect protection \* AMBA external interface protection \* Q-Channel protection \* Systematic fault watchdog \* Clock and reset duplication

**Iris and MTI instances for MHU320AE**

This model has the following Iris instances:

Name	Instance type
MHU320AE	MHU320AE
MHU320AE.MHU320AE_FMU	mhu320ae_fmu

This model has the following MTI trace components:

Name	Component type
MHU320AE	MessageHandlingUnitV3

## Ports for MHU320AE

Port	Direction	Protocol	Description
fmu_cri_out	master	Signal	-
fmu_eri_out	master	Signal	-
fmu_reset_in	slave	Signal	-
pvbus_s_rec	slave	PVBus	Register access for Receiver/Mailbox
pvbus_s_snd	slave	PVBus	Register access for Sender/Postbox
rec_combined_irq_out	master	Signal	All interrupts combined for Receiver/MBX
rec_fast_channel_group_irq_out	master	Signal	Receiver fast channel group interrupts
rec_fast_channel_irq_out	master	Signal	Receiver fast channel interrupts
rec_reset_in	slave	Signal	Reset signal for Receiver/Mailbox
recv_fmu_pvbus_s	slave	PVBus	-
send_fmu_pvbus_s	slave	PVBus	FUSA related FMU signals
snd_combined_irq_out	master	Signal	All Interrupts combined for Sender/PBX
snd_reset_in	slave	Signal	Reset signal for Sender/Postbox

## Parameters for MHU320AE

### NUM\_DB\_CH

Number of doorbell channels.

Type: uint32\_t

Default value: 1

### NUM\_FAST\_CH

Number of Fast Channels.

Type: uint32\_t

Default value: 1

### NUM\_FIFO\_CH

Number of FIFO Channels.



Type: `uint32_t`

Default value: 1

### **auto\_op\_full**

AutoOp mode - AutoOp(min) == false, AutoOp(full) == true - default: AutoOp(min).

Type: `bool`

Default value: `false`

### **diagnostics**

Diagnostics 0:FATAL 1:ERROR 2:WARNING 3:INFO 4:DEBUG, Default:WARNING(2).

Type: `uint8_t`

Default value: 2

### **fast\_ch\_group\_int\_enable**

Fast Channel group interrupts enable, default=false.

Type: `bool`

Default value: `false`

### **fast\_ch\_n\_per\_group**

Fast Channel num channels per group, default=1.

Type: `uint32_t`

Default value: 1

### **fast\_ch\_num\_groups**

Fast Channel num of groups, default=1.

Type: `uint32_t`

Default value: 1

### **fast\_ch\_word\_size**

Fast Channel word size 32bit or 64bit, default=32.

Type: `uint32_t`

Default value: 32

### **fifo\_depth**

Depth of the FIFO = `fifo_depth` + 1.

Type: `uint16_t`

Default value: 4

### **fmu\_location**

FMU LOCATION: 0-2 (0:SENDER 1:RECEIVER 2:BOTH).

Type: `uint32_t`

Default value: 0

### **m16ba\_spt**

Mailbox 16 bit access support to FIFO registers.

Type: `"bool"`

Default value: 0

### **m32ba\_spt**

Mailbox 32 bit access support to FIFO registers.

Type: `"bool"`

Default value: 1

### **m64ba\_spt**

Mailbox 64 bit access support to FIFO registers.

Type: `"bool"`

Default value: 0

### **m8ba\_spt**

Mailbox 8 bit access support to FIFO registers.

Type: `"bool"`

Default value: 0

### **monolithic**

Monolithic or Distributed MHU - default: `monolithic(true)`.

Type: `bool`

Default value: `true`

### **p16ba\_spt**

Postbox 16 bit access support to FIFO registers.

Type: "bool"

Default value: 0

**p32ba\_spt**

Postbox 32 bit access support to FIFO registers.

Type: "bool"

Default value: 1

**p64ba\_spt**

Postbox 64 bit access support to FIFO registers.

Type: "bool"

Default value: 0

**p8ba\_spt**

Postbox 8 bit access support to FIFO registers.

Type: "bool"

Default value: 0

3.238 MMC

Defined in LISA/MMC.lisa.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

About MMC

This component simulates an SD or SDHC card that is compatible with the [MultiMedia Card Association, MMCA](#) specification version 3.31. The parameters permit configuration of a number of attributes reflected in the CID and CSD registers. You can customize the component further by modifying the supplied MMC model source code directly.

When paired with a PL180\_MCI component, the MMC device model provides emulation of a flexible, persistent storage mechanism.

The MMC component uses a file on the host PC to simulate the storage device. The size of this backing store file determines the reported size of the MMC device. As small sections of this file are paged in by the model, large filesystems can be modeled while making efficient use of host PC memory. The backing store file can contain a partition table and filesystems such as FAT or EXT2.

The image file is a direct bit copy of the contents of an SD card. If the image file that the `p_mmc_file` parameter refers to does not exist, the component behaves as if the card is absent. If the image file is read-only, the component behaves as if the card is read-only.



Operating system boots often attempt to write to the boot filesystem. They might not work properly if the boot filesystem is on a read-only card.

---

The MMC component does not model card insertion or removal. It models the card having already been inserted at system instantiation time.

You can configure the MMC component to behave as an SDHC card by setting the `card_type` parameter to `SDHC`. SDHC mode is a model-specific extension, and is not supported by PL180 hardware. It supports filesystems that are larger than 2 GB.

The component supports these commands:

- `MMC_GO_IDLE_STATE`
- `MMC_SEND_OP_COND`
- `MMC_ALL_SEND_CID`
- `MMC_SET_RELATIVE_ADDR`
- `MMC_SET_DSR`
- `MMC_SELDESL_CARD`
- `MMC_SEND_CSD`
- `MMC_SEND_CID`
- `MMC_STOP_TRANSMISSION`
- `MMC_SEND_STATUS`
- `MMC_GO_INACTIVE_STATE`
- `MMC_READ_SINGLE_BLOCK`
- `MMC_READ_MULTIPLE_BLOCK`
- `MMC_SET_BLOCK_COUNT`
- `MMC_WRITE_BLOCK`
- `MMC_WRITE_MULTIPLE_BLOCK`
- `MMC_SEND_EXT_CSD`. This command is supported in SDHC mode only

The block length is 512 bytes. SimGen reports attempts to change it as errors.

The component supports these erase commands (Class 5), but they have no effect on the disk backing storage:

- `MMC_ERASE_GROUP_START`
- `MMC_ERASE_GROUP_END`
- `MMC_ERASE`

The component does not support these commands:

- `MMC_BUSTEST_R`
- `MMC_BUSTEST_W`

The component does not support stream read and write commands (Classes 1 and 3):

- `MMC_READ_DAT_UNTIL_STOP`
- `MMC_WRITE_DAT_UNTIL_STOP`
- `MMC_PROGRAM_CID`
- `MMC_PROGRAM_CSD`

The component does not support block oriented write protection commands (Class 6):

- `MMC_SET_WRITE_PROT`
- `MMC_CLR_WRITE_PROT`
- `MMC_SEND_WRITE_PROT`

The component does not support lock card commands (Class 7) or application-specific commands (Class 8):

- `MMC_LOCK_UNLOCK`
- `MMC_APP_CMD`
- `MMC_GEN_CMD`

The component does not support I/O mode commands (Class 9):

- `MMC_FAST_IO`
- `MMC_GO_IRQ_STATE`

The component does not support reserved commands. Using a reserved command sets the `MMC_ST_ER_B_ILLEGAL_COMMAND` bit in the status register of the card. Read this with the `MMC_SEND_STATUS` command.

Use the `p_diagnostics` parameter to select the level of diagnostic output, to help to debug device driver and controller-to-card protocol issues. It supports the following levels:

### Level 0

None

**Level 1**

Warnings about attempting to change read-only settings.

**Level 2**

Trace of command calls.

**Level 3**

Information about every step in the MMC\_Protocol interaction.

**Level 4**

Hex dump of every block sent or received.

The registers are not memory mapped. Instead, you access them using relevant MMC commands. The MMC component model makes the registers available through an Iris interface. Modification of these registers through Iris is not recommended, but not prohibited. For example, modifying the card ID (CID) registers can be useful when experimenting with drivers, but direct modification of the STATUS\_REG register is likely to put the card model into an indeterminate state.

For a full definition of MMC registers, see the MMCA System Summary documentation. Device-specific register information can also be obtained from MMC vendors.

**Table 3-820: MMC registers**

Name	Iris register number	Description
OCR_REG	0x000	Operating conditions register
CID_REG0	0x004	Card ID bits 127:96
CID_REG1	0x005	Card ID bits 95:64
CID_REG2	0x006	Card ID bits 63:32
CID_REG3	0x007	Card ID bits 31:0
CSD_REG0	0x008	Card specific data bits 127:96
CSD_REG1	0x009	Card specific data bits 95:64
CSD_REG2	0x00a	Card specific data bits 63:32
CSD_REG3	0x00b	Card specific data bit 31:0
RCA_REG	0x00c	Relative card address register
DSR_REG	0x00d	Driver stage register
BLOCKLEN_REG	0x00e	Block length
STATUS_REG	0x00f	Card status
BLOCK_COUNT_REG	0x010	Block count

**Iris and MTI instances for MMC**

This model has the following Iris instances:

Name	Instance type
MMC	MMC
MMC.timer	<a href="#">ClockTimerThread</a>
MMC.timer.timer	<a href="#">ClockTimerThread64</a>
MMC.timer.timer.thread	<a href="#">SchedulerThread</a>

Name	Instance type
<code>MMC.timer.timer.thread_event</code>	<code>SchedulerThreadEvent</code>

## Ports for MMC

Port	Direction	Protocol	Description
<code>card_present</code>	master	<code>StateSignal</code>	-
<code>clk_in</code>	slave	<code>ClockSignal</code>	-
<code>mmc</code>	slave	<code>MMC_Protocol</code>	-

## Parameters for MMC

### **card\_type**

Card type('SD' or 'SDHC'.

Type: `string`

Default value: `"SDHC"`

### **diagnostics**

Diagnostics level.

Type: `int`

Default value: `0`

### **force\_sector\_addressing**

Use sector addressing even on small cards.

Type: `bool`

Default value: `false`

### **p\_OEMid**

Card ID OEM ID.

Type: `int`

Default value: `0x0000`

### **p\_fast\_access**

Don't simulate MMC block access delays.

Type: `bool`

Default value: `true`

**p\_manid**

Card ID Manufacturer ID.

Type: `int`

Default value: `0x02`

**p\_max\_block\_count**

Default maximum block count reg. Default `0x80`.

Type: `uint32_t`

Default value: `0x80`

**p\_mmc\_file**

MMCard filename.

Type: `string`

Default value: `"mmc.dat"`

**p\_prodName**

Card ID Product Name (6 chars).

Type: `string`

Default value: `"ARMmmc"`

**p\_prodRev**

Card ID Product Revision.

Type: `int`

Default value: `0x01`

**p\_sernum**

Card Serial Number.

Type: `int`

Default value: `0xca4d0001`

**support\_unpadded\_images**

Support images that are not a multiple of 512k by padding them to the needed size (SDHC cards only).

Type: `bool`



Default value: `false`

## 3.239 MMU\_400

Defined in `LISA/SMMU_400.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### About MMU\_400

This provides a basic MMU-400 that is configurable. It does not allow arbitrary configuration with respect to how StreamIDs and SSD\_Indexes are derived from the transaction attributes.

Set the `use_label_mapping` parameter to `true` if your upstream devices have labels in the [31:16] bits of the transaction ManagerID.



The model does not have a concept of AXI-ID, but a transaction can have a ManagerID set on it.

Label your upstream components `0...N` so that the parameters of this component can map those integers to StreamID and SSD\_Index.

Set `use_label_mapping` to `false` if the StreamID is encoded in the top 16 bits of the ManagerID and the bottom 16 bits encode either the SSD\_Index or the SSD state directly, depending on `use_ssd_determination_table`. Typically in hardware, a device emits different AXI-IDs, depending on what it is doing. In the model, ManagerIDs are usually not diverse and a device might only emit one ManagerID.

If `use_ssd_determination_table` is `true`, the bottom 16 bits of the ManagerID encode the SSD\_Index. They must be  $< 2^{\text{ssd\_index\_width}}$ . If it is `false`, they encode the SSD state directly (zero is Secure and nonzero is Non-secure).

This component models all architectural registers that are specified in the Technical Reference Manual (TRM), except that it does not model any of the performance registers, and has the following limitations:

- MMU-400 does not have an SMMU\_STLBGSTATUS register because the Secure side is a nominal pass-through. MMU-400 only has stage 2 support and you cannot use stage 2 on the Secure side.

- The SMMU\_NSACR is an alias of the Non-secure SMMU\_ACR. This component models SMMU\_ACR as **RAZ/WI**.
- The \*ACR registers have IMP DEF contents. This component models only the PAGESIZE bit of the SACR, as non-**RAZ/WI**. It models no other IMP DEF registers

### Iris and MTI instances for MMU\_400

This model has the following Iris instances:

Name	Instance type
MMU_400	MMU_400
MMU_400.mmu	MMU_400_BASE
MMU_400.mmu.apb3_control_ns_slv	PVBusSlave
MMU_400.mmu.apb3_control_s_slv	PVBusSlave
MMU_400.mmu.apb4_control_slv	PVBusSlave
MMU_400.mmu.mapper	PVBusMapper
MMU_400.mmu.ptw_dvm_receiver	PVBusMapper
MMU_400.mmu.ptw_master	PVBusMaster
MMU_400.mmu.pvbus_master	PVBusMaster

This model has the following MTI trace components:

Name	Component type
MMU_400.mmu	MMU_400_BASE
MMU_400.mmu.apb3_control_ns_slv	PVBusSlave
MMU_400.mmu.apb3_control_s_slv	PVBusSlave
MMU_400.mmu.apb4_control_slv	PVBusSlave
MMU_400.mmu.mapper	PVBusMapper
MMU_400.mmu.ptw_dvm_receiver	PVBusMapper
MMU_400.mmu.ptw_master	PVBusMaster
MMU_400.mmu.pvbus_master	PVBusMaster

### Ports for MMU\_400

Port	Direction	Protocol	Description
apb3_control_ns	slave	PVBus	APBv3 control port for Non-secure access to the register file. If this port is used do not use the APBv4 port.
apb3_control_s	slave	PVBus	APBv3 control port for Secure access to the register file. If this port is used do not use the APBv4 port.
apb4_control	slave	PVBus	APBv4 control port for access to the register file. If this port is used do not use the APBv3 ports.
cfg_cttw_in	slave	Signal	Enables coherent page table walks.
cfg_flt_irpt_ns	master	Signal	Non-secure configuration access fault interrupt. Corresponds to SMMU architectural signal SMMU_NSgCfgrpt.
cfg_flt_irpt_s	master	Signal	Secure configuration access fault interrupt. Corresponds to SMMU architectural signal SMMU_gCfgrpt.

Port	Direction	Protocol	Description
comb_irpt_ns	master	Signal	Non-secure combined interrupt.
comb_irpt_s	master	Signal	Secure combined interrupt.
cxt_irpt_ns	master	Signal	Non-secure context bank fault.
glblflt_irpt_ns	master	Signal	Global Non-secure fault interrupt. Corresponds to SMMU architectural signal SMMU_NSglrpt.
glblflt_irpt_s	master	Signal	Global Secure fault interrupt. Corresponds to SMMU architectural signal SMMU_glrpt.
priv_internals	slave	MMU_400_Internals	For internal use only, please do not use.
pvbustm	master	PVBus	Downstream port of the MMU, where translated transactions emerge.
pvbustptw_tm	master	PVBus	Downstream port for page table walks if configured using the ptw_has_separate_port parameter.
pvbusts	slave	PVBus	Upstream port of the MMU. Addresses on the port are in VA/IPA.
reset_in	slave	Signal	Signal to reset the MMU.

## Parameters for MMU\_400

### **always\_secure\_ssd\_indices**

Non-programmable SSD Indexes that are always secure (e.g. 0, 6, 35-64).

Type: `string`

Default value: `""`

### **cfg\_cttw**

Perform coherent page table walks.

Type: `bool`

Default value: `true`

### **dump\_unpredictability\_in\_user\_flags**

Override the user flags to encode unpredictable information (validation only).

Type: `bool`

Default value: `false`

### **label0\_read\_ssd**

Label0: Read SDD or SSD\_Index.

Type: `unsigned`

Default value: `0`

**label0\_read\_stream\_id**

Label0: Read Stream ID.

Type: unsigned

Default value: 0

**label0\_write\_ssd**

Label0: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label0\_write\_stream\_id**

Label0: Write Stream ID.

Type: unsigned

Default value: 0

**label10\_read\_ssd**

Label10: Read SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label10\_read\_stream\_id**

Label10: Read Stream ID.

Type: unsigned

Default value: 0

**label10\_write\_ssd**

Label10: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label10\_write\_stream\_id**

Label10: Write Stream ID.

Type: unsigned

Default value: 0

**label11\_read\_ssd**

Label11: Read SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label11\_read\_stream\_id**

Label11: Read Stream ID.

Type: unsigned

Default value: 0

**label11\_write\_ssd**

Label11: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label11\_write\_stream\_id**

Label11: Write Stream ID.

Type: unsigned

Default value: 0

**label12\_read\_ssd**

Label12: Read SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label12\_read\_stream\_id**

Label12: Read Stream ID.

Type: unsigned

Default value: 0

**label12\_write\_ssd**

Label12: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label12\_write\_stream\_id**

Label12: Write Stream ID.

Type: unsigned

Default value: 0

**label13\_read\_ssd**

Label13: Read SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label13\_read\_stream\_id**

Label13: Read Stream ID.

Type: unsigned

Default value: 0

**label13\_write\_ssd**

Label13: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label13\_write\_stream\_id**

Label13: Write Stream ID.

Type: unsigned

Default value: 0

**label14\_read\_ssd**

Label14: Read SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label14\_read\_stream\_id**

Label14: Read Stream ID.

Type: unsigned

Default value: 0

**label14\_write\_ssd**

Label14: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label14\_write\_stream\_id**

Label14: Write Stream ID.

Type: unsigned

Default value: 0

**label15\_read\_ssd**

Label15: Read SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label15\_read\_stream\_id**

Label15: Read Stream ID.

Type: unsigned

Default value: 0

**label15\_write\_ssd**

Label15: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label15\_write\_stream\_id**

Label15: Write Stream ID.

Type: unsigned

Default value: 0

**label16\_read\_ssd**

Label16: Read SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label16\_read\_stream\_id**

Label16: Read Stream ID.

Type: unsigned

Default value: 0

**label16\_write\_ssd**

Label16: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label16\_write\_stream\_id**

Label16: Write Stream ID.

Type: unsigned

Default value: 0

**label17\_read\_ssd**

Label17: Read SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label17\_read\_stream\_id**

Label17: Read Stream ID.

Type: unsigned

Default value: 0

**label17\_write\_ssd**

Label17: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label17\_write\_stream\_id**

Label17: Write Stream ID.

Type: unsigned

Default value: 0



**label18\_read\_ssd**

Label18: Read SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label18\_read\_stream\_id**

Label18: Read Stream ID.

Type: unsigned

Default value: 0

**label18\_write\_ssd**

Label18: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label18\_write\_stream\_id**

Label18: Write Stream ID.

Type: unsigned

Default value: 0

**label19\_read\_ssd**

Label19: Read SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label19\_read\_stream\_id**

Label19: Read Stream ID.

Type: unsigned

Default value: 0

**label19\_write\_ssd**

Label19: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label19\_write\_stream\_id**

Label19: Write Stream ID.

Type: unsigned

Default value: 0

**label1\_read\_ssd**

Label1: Read SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label1\_read\_stream\_id**

Label1: Read Stream ID.

Type: unsigned

Default value: 0

**label1\_write\_ssd**

Label1: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label1\_write\_stream\_id**

Label1: Write Stream ID.

Type: unsigned

Default value: 0

**label20\_read\_ssd**

Label20: Read SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label20\_read\_stream\_id**

Label20: Read Stream ID.

Type: unsigned

Default value: 0

**label20\_write\_ssd**

Label20: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label20\_write\_stream\_id**

Label20: Write Stream ID.

Type: unsigned

Default value: 0

**label21\_read\_ssd**

Label21: Read SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label21\_read\_stream\_id**

Label21: Read Stream ID.

Type: unsigned

Default value: 0

**label21\_write\_ssd**

Label21: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label21\_write\_stream\_id**

Label21: Write Stream ID.

Type: unsigned

Default value: 0

**label22\_read\_ssd**

Label22: Read SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label122\_read\_stream\_id**

Label22: Read Stream ID.

Type: unsigned

Default value: 0

**label122\_write\_ssd**

Label22: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label122\_write\_stream\_id**

Label22: Write Stream ID.

Type: unsigned

Default value: 0

**label123\_read\_ssd**

Label23: Read SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label123\_read\_stream\_id**

Label23: Read Stream ID.

Type: unsigned

Default value: 0

**label123\_write\_ssd**

Label23: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label123\_write\_stream\_id**

Label23: Write Stream ID.

Type: unsigned

Default value: 0

**label124\_read\_ssd**

Label24: Read SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label124\_read\_stream\_id**

Label24: Read Stream ID.

Type: unsigned

Default value: 0

**label124\_write\_ssd**

Label24: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label124\_write\_stream\_id**

Label24: Write Stream ID.

Type: unsigned

Default value: 0

**label125\_read\_ssd**

Label25: Read SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label125\_read\_stream\_id**

Label25: Read Stream ID.

Type: unsigned

Default value: 0

**label125\_write\_ssd**

Label25: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label125\_write\_stream\_id**

Label25: Write Stream ID.

Type: unsigned

Default value: 0

**label126\_read\_ssd**

Label26: Read SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label126\_read\_stream\_id**

Label26: Read Stream ID.

Type: unsigned

Default value: 0

**label126\_write\_ssd**

Label26: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label126\_write\_stream\_id**

Label26: Write Stream ID.

Type: unsigned

Default value: 0

**label127\_read\_ssd**

Label27: Read SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label127\_read\_stream\_id**

Label27: Read Stream ID.

Type: unsigned

Default value: 0

**label27\_write\_ssd**

Label27: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label27\_write\_stream\_id**

Label27: Write Stream ID.

Type: unsigned

Default value: 0

**label28\_read\_ssd**

Label28: Read SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label28\_read\_stream\_id**

Label28: Read Stream ID.

Type: unsigned

Default value: 0

**label28\_write\_ssd**

Label28: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label28\_write\_stream\_id**

Label28: Write Stream ID.

Type: unsigned

Default value: 0

**label29\_read\_ssd**

Label29: Read SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label29\_read\_stream\_id**

Label29: Read Stream ID.

Type: unsigned

Default value: 0

**label29\_write\_ssd**

Label29: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label29\_write\_stream\_id**

Label29: Write Stream ID.

Type: unsigned

Default value: 0

**label2\_read\_ssd**

Label2: Read SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label2\_read\_stream\_id**

Label2: Read Stream ID.

Type: unsigned

Default value: 0

**label2\_write\_ssd**

Label2: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label2\_write\_stream\_id**

Label2: Write Stream ID.

Type: unsigned

Default value: 0



**label130\_read\_ssd**

Label130: Read SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label130\_read\_stream\_id**

Label130: Read Stream ID.

Type: unsigned

Default value: 0

**label130\_write\_ssd**

Label130: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label130\_write\_stream\_id**

Label130: Write Stream ID.

Type: unsigned

Default value: 0

**label131\_read\_ssd**

Label131: Read SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label131\_read\_stream\_id**

Label131: Read Stream ID.

Type: unsigned

Default value: 0

**label131\_write\_ssd**

Label131: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label31\_write\_stream\_id**

Label31: Write Stream ID.

Type: unsigned

Default value: 0

**label3\_read\_ssd**

Label3: Read SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label3\_read\_stream\_id**

Label3: Read Stream ID.

Type: unsigned

Default value: 0

**label3\_write\_ssd**

Label3: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label3\_write\_stream\_id**

Label3: Write Stream ID.

Type: unsigned

Default value: 0

**label4\_read\_ssd**

Label4: Read SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label4\_read\_stream\_id**

Label4: Read Stream ID.

Type: unsigned

Default value: 0

**label4\_write\_ssd**

Label4: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label4\_write\_stream\_id**

Label4: Write Stream ID.

Type: unsigned

Default value: 0

**label5\_read\_ssd**

Label5: Read SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label5\_read\_stream\_id**

Label5: Read Stream ID.

Type: unsigned

Default value: 0

**label5\_write\_ssd**

Label5: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label5\_write\_stream\_id**

Label5: Write Stream ID.

Type: unsigned

Default value: 0

**label6\_read\_ssd**

Label6: Read SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label6\_read\_stream\_id**

Label6: Read Stream ID.

Type: unsigned

Default value: 0

**label6\_write\_ssd**

Label6: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label6\_write\_stream\_id**

Label6: Write Stream ID.

Type: unsigned

Default value: 0

**label7\_read\_ssd**

Label7: Read SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label7\_read\_stream\_id**

Label7: Read Stream ID.

Type: unsigned

Default value: 0

**label7\_write\_ssd**

Label7: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label7\_write\_stream\_id**

Label7: Write Stream ID.

Type: unsigned

Default value: 0

**label8\_read\_ssd**

Label8: Read SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label8\_read\_stream\_id**

Label8: Read Stream ID.

Type: unsigned

Default value: 0

**label8\_write\_ssd**

Label8: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label8\_write\_stream\_id**

Label8: Write Stream ID.

Type: unsigned

Default value: 0

**label9\_read\_ssd**

Label9: Read SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label9\_read\_stream\_id**

Label9: Read Stream ID.

Type: unsigned

Default value: 0

**label9\_write\_ssd**

Label9: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label9\_write\_stream\_id**

Label9: Write Stream ID.

Type: unsigned

Default value: 0

**number\_of\_contexts**

Number of context banks.

Type: unsigned

Default value: 8

**number\_of\_smrs**

Number of stream match registers.

Type: unsigned

Default value: 32

**percent\_tlbstatus\_commits**

Percentage of times that a poll of TLBSTATUS will commit the TLBI commands.

Type: uint32\_t

Default value: 10

**prefetch\_only\_requests**

Handle prefetch-only requests by:- – deny them – use debug table walks and TLB entries – treat them as normal transactions (dangerous).

Type: unsigned

Default value: 0

**programmable\_non\_secure\_by\_default\_ssd\_indices**

Programmable SSD Indexes that are by default non-secure (e.g. 0, 6, 35-84).

Type: string

Default value: ""

**programmable\_secure\_by\_default\_ssd\_indices**

Programmable SSD Indexes that are by default secure (e.g. 0, 6, 35-84).

Type: string

Default value: ""

**ptw\_has\_separate\_port**

Page Table Walks use pvbus\_ptw\_m.

Type: bool

Default value: true

**pvbus\_m\_is\_ace\_lite**

Is pvbus\_m (the downstream port that translated transaction exit) ACE-Lite.

Type: bool

Default value: true

**pvbus\_ptw\_m\_is\_ace\_lite**

Is pvbus\_ptw\_m (the downstream port that is used for walks if ptw\_has\_separate\_port is true) ACE-Lite.

Type: bool

Default value: true

**stream\_id\_width**

StreamID bit width.

Type: unsigned

Default value: 6

**tlb\_depth**

TLB Depth (0 means 10000). The model will perform best with more TLB entries.

Type: unsigned

Default value: 64

**use\_label\_mapping**

Use label mapping.

Type: bool

Default value: true

**use\_ssd\_determination\_table**

Use SSD Determination Table.

Type: `bool`

Default value: `true`

## 3.240 MMU\_400\_BASE

Defined in `LISA/SMMU_400_BASE.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### About MMU\_400\_BASE

MMU-400 base component.

### Iris and MTI instances for MMU\_400\_BASE

This model has the following Iris instances:

Name	Instance type
MMU_400_BASE	MMU_400_BASE
MMU_400_BASE.apb3_control_ns_slv	PVBusSlave
MMU_400_BASE.apb3_control_s_slv	PVBusSlave
MMU_400_BASE.apb4_control_slv	PVBusSlave
MMU_400_BASE.mapper	PVBusMapper
MMU_400_BASE.ptw_dvm_receiver	PVBusMapper
MMU_400_BASE.ptw_master	PVBusMaster
MMU_400_BASE.pvbus_master	PVBusMaster

This model has the following MTI trace components:

Name	Component type
MMU_400_BASE	MMU_400_BASE
MMU_400_BASE.apb3_control_ns_slv	PVBusSlave
MMU_400_BASE.apb3_control_s_slv	PVBusSlave
MMU_400_BASE.apb4_control_slv	PVBusSlave
MMU_400_BASE.mapper	PVBusMapper
MMU_400_BASE.ptw_dvm_receiver	PVBusMapper
MMU_400_BASE.ptw_master	PVBusMaster
MMU_400_BASE.pvbus_master	PVBusMaster



## Ports for MMU\_400\_BASE

Port	Direction	Protocol	Description
apb3_control_ns	slave	PVBus	If the device has been configured with APB3 control ports then this is used to address the register file with non-secure accesses. If this is the case then the apb4_control port should not be used.
apb3_control_s	slave	PVBus	If the device has been configured with APB3 control ports then this is used to address the register file with secure accesses. If this is the case then the apb4_control port should not be used.
apb4_control	slave	PVBus	If the device has been configured with APB4 control ports then this port is used – it carries the security world with the transaction itself. If this is the case then the apb3_control_s and apb3_control_ns should not be used.
cfg_cttw_in	slave	Signal	The SoC supports coherent page walks, this is meant to be sampled at reset. However, in practice the model has to prevent the race condition between cfg_cttw being asserted at the same 'cycle' as negedge reset. Thus we actually only sample the signal on the first transaction to the SMMU or the first transition on this signal after reset. Thus in the model, we require that cfg_cttw be held for at least this period of time.
cfgflt_irpt_ns	master	Signal	Non-secure Configuration Access Fault Interrupt In the SMMU Architecture this is called SMMU_NSgCfgrpt.
cfgflt_irpt_s	master	Signal	Secure Configuration Access Fault Interrupt In the SMMU Architecture this is called SMMU_gCfgrpt.
comb_irpt_ns	master	Signal	“Non-secure combined interrupt” (cfgflt_irpt_ns   glblflt_irpt_ns   cxt_irpt_ns)?
comb_irpt_s	master	Signal	“Secure combined interrupt”
cxt_irpt_ns	master	Signal	Non-secure context bank fault NOTE that there is only one context bank fault, despite there being potentially 8 contexts. As we are HW stage 2 only then we can't have any banks configured as secure (well if we do then we generate a global fault).
glblflt_irpt_ns	master	Signal	Global non-secure Fault Interrupt In the SMMU Architecture this is called SMMU_NSgIrpt.
glblflt_irpt_s	master	Signal	Global secure Fault Interrupt In the SMMU Architecture this is called SMMU_gIrpt.
identify	master	MMU_400_BASE_IDENTIFY	This port is a special model port that is used to take a transaction and map it to an SSD/SSD_Index and StreamID.
priv_internals	slave	MMU_400_Internals	For internal use only, please do not use.
pvbus_m	master	PVBus	This downstream port is where the translated accesses from pvbus_s emerge. If page walks are configured to come out of this port, then they will come out with the with the same attributes as described for pvbus_ptw_m.

Port	Direction	Protocol	Description
pvbus_ptw_m	master	PVBus	This downstream port is where page table walk accesses come from. This is only used if configured to use a separate page table walk port. The MMU-400 will only obey DVM messages if configured to use this port. The page walks come out of this port with the following manager_id and user_flags. manager_id : 0xFFFFfff The user flags : user_flags[7:0] stage 1 context_id (or 0xFF if stage2 only) user_flags[15:8] stage 2 context_id (or 0xFF if stage 1 with stage 2 bypass) user_flags[18:16] stage 1 level user_flags[21:19] stage 2 level user_flags[23:22] stage 1 descriptor encoding (0=v7s, 1=v7l, 2=v8l, 3=none) user_flags[25:24] stage 2 descriptor encoding (0=v7s, 1=v7l, 2=v8l, 3=none) user_flags[31,30] adomain of the transaction NOTE that if the walk is being done for a stage 1 page walk descriptor fetch then the stage 1 level field will indicate that level. If the walk is being done for a stage 2 descriptor fetch, then the stage 2 level field will show that level. If the context-id for a stage is not valid (0xFF) then the 'level' information is 0x7.
pvbus_s	slave	PVBus	This port is the upstream port of the device, addresses on the port are in the VA/IPA
reset_in	slave	Signal	The reset pin.

## Parameters for MMU\_400\_BASE

### **always\_secure\_ssd\_indices**

Non-programmable SSD Indexes that are always secure (e.g. 0, 6, 35-64).

Type: string

Default value: ""

### **cfg\_cttw**

Perform coherent page table walks.

Type: bool

Default value: true

### **dump\_unpredictablity\_in\_user\_flags**

Override the user flags to encode unpredictable information (validation only).

Type: bool

Default value: false

### **number\_of\_contexts**

Number of context banks.

Type: unsigned

Default value: 8

**number\_of\_smrs**

Number of stream match registers.

Type: unsigned

Default value: 16

**percent\_tlbstatus\_commits**

Percentage of times that a poll of TLBSTATUS will commit the TLBI commands.

Type: uint32\_t

Default value: 10

**prefetch\_only\_requests**

Handle prefetch-only requests by:- – deny them – use debug table walks and TLB entries – treat them as normal transactions (dangerous).

Type: unsigned

Default value: 0

**programmable\_non\_secure\_by\_default\_ssd\_indices**

Programmable SSD Indexes that are by default non-secure (e.g. 0, 6, 35-84).

Type: string

Default value: ""

**programmable\_secure\_by\_default\_ssd\_indices**

Programmable SSD Indexes that are by default secure (e.g. 0, 6, 35-84).

Type: string

Default value: ""

**ptw\_has\_separate\_port**

Page Table Walks use pvbus\_ptw\_m.

Type: bool

Default value: true

**pvbus\_m\_is\_ace\_lite**

Is pvbus\_m (the downstream port that translated transaction exit) ACE-Lite.

Type: bool

Default value: `true`

**pvbus\_ptw\_m\_is\_ace\_lite**

Is pvbus\_ptw\_m (the downstream port that is used for walks if ptw\_has\_separate\_port is true) ACE-Lite.

Type: `bool`

Default value: `true`

**seed**

Seed for SMMU.

Type: `uint32_t`

Default value: `0x12345678`

**stream\_id\_width**

StreamID bit width.

Type: `unsigned`

Default value: `6`

**tlb\_depth**

TLB Depth (0 means 10000). The model will perform best with more TLB entries.

Type: `unsigned`

Default value: `64`

**use\_ssd\_determination\_table**

Use SSD Determination Table.

Type: `bool`

Default value: `true`

3.241 MMU\_500

Defined in `LISA/SMMU_500.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r2p4	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

## About MMU\_500

This is a model of a basic MMU-500. Set the version using the `version` parameter.

You cannot arbitrarily configure how you derive StreamIDs and SSD\_Indexes from the transaction attributes.

This component has two label modes which you select using the parameter `use_label_mapping`:

- Set `use_label_mapping` to `true` if your upstream devices have labels in the [31:16] bits of the transaction ManagerID.



The model does not have a concept of AXI-ID, but a transaction can have a ManagerID set on it.

Label your upstream components  $0 \dots N$  so that the parameters of this component can map those integers to StreamID and SSD\_Index.

- Set `use_label_mapping` to `false` if the StreamID is encoded in the top 16 bits of the ManagerID and the bottom 16 bits encode either the SSD\_Index or the SSD state directly, depending on `use_ssd_determination_table`:
  - If `use_ssd_determination_table` is `true`, the bottom 16 bits of the ManagerID encode the SSD\_Index. They must be  $< 2^{\text{ssd\_index\_width}}$ .
  - If `use_ssd_determination_table` is `false`, the bottom 16 bits of the ManagerID encode the SSD state directly, where zero is Secure and nonzero is Non-secure.

Typically in hardware, a device emits different AXI-IDs, depending on what it is doing. In the model, ManagerIDs are usually not diverse and a device might only emit one ManagerID.

This component models the registers as follows:

- It models all architectural registers that the Technical Reference Manual (TRM) specifies, except that it does not model any of the performance registers.
- Unlike the MMU-400, MMU-500 does have an SMMU\_STLBGSTATUS register because it has stage 1 and stage 2 support.
- The SMMU\_NSACR is an alias of the Non-secure SMMU\_ACR. This component models SMMU\_ACR as **RAZ/WI**.
- The \*ACR registers have IMP DEF contents. This component models only the PAGESIZE bit of the SACR, as non-**RAZ/WI**. It models no other IMP DEF registers.

## Iris and MTI instances for MMU\_500

This model has the following Iris instances:

Name	Instance type
MMU_500	MMU_500

Name	Instance type
MMU_500.mmu	MMU_500_BASE
MMU_500.mmu.mapperY (where Y = 0-31)	PVBusMapper
MMU_500.mmu.ptw_dvm_receiver	PVBusMapper
MMU_500.mmu.ptw_master	PVBusMaster
MMU_500.mmu.pvbus_control_s_slv	PVBusSlave
MMU_500.mmu.pvbus_master	PVBusMaster
MMU_500.mmu.pvbusmasterY (where Y = 0-31)	PVBusMaster
MMU_500.mmu.pvbusslaveY (where Y = 0-31)	PVBusSlave

This model has the following MTI trace components:

Name	Component type
MMU_500.mmu.mapperY (where Y = 0-31)	PVBusMapper
MMU_500.mmu.ptw_dvm_receiver	PVBusMapper
MMU_500.mmu.ptw_master	PVBusMaster
MMU_500.mmu.pvbus_control_s_slv	PVBusSlave
MMU_500.mmu.pvbus_master	PVBusMaster
MMU_500.mmu.pvbusmasterY (where Y = 0-31)	PVBusMaster
MMU_500.mmu.pvbusslaveY (where Y = 0-31)	PVBusSlave

## Ports for MMU\_500

Port	Direction	Protocol	Description
cfg_cttw_in	slave	Signal	Enables coherent page table walks.
comb_irpt_ns	master	Signal	Non-secure combined interrupt.
comb_irpt_s	master	Signal	Secure combined interrupt.
cxt_irpt	master	Signal	Context interrupt.
glblflt_irpt_ns	master	Signal	Global Non-secure fault interrupt.
glblflt_irpt_s	master	Signal	Global Secure fault interrupt.
priv_internals	slave	MMU_500_Internals	For internal use only, please do not use.
pvbus_control_s	slave	PVBus	Provides memory-mapped read write access to the control registers of the module.
pvbus_m	master	PVBus	For all memory accesses. One for each Translation Buffer Unit (TBU).
pvbus_ptw_m	master	PVBus	If ptw_has_separate_port is true, use for page table walks.
pvbus_s	slave	PVBus	For transactions from PVBus master/decoder. One for each TBU.
reset_in	slave	Signal	Reset signal.

## Parameters for MMU\_500

### **always\_secure\_ssd\_indices**

Non-programmable SSD Indexes that are always secure (e.g. 0, 6, 35-64).

Type: string

Default value: ""

**cfg\_cttw**

Perform coherent page table walks.

Type: `bool`

Default value: `true`

**dump\_unpredictability\_in\_user\_flags**

Override the user flags to encode unpredictable information (validation only).

Type: `bool`

Default value: `false`

**label0\_read\_ssd**

Label0: Read SDD or SSD\_Index.

Type: `unsigned`

Default value: 0

**label0\_read\_stream\_id**

Label0: Read Stream ID.

Type: `unsigned`

Default value: 0

**label0\_write\_ssd**

Label0: Write SDD or SSD\_Index.

Type: `unsigned`

Default value: 0

**label0\_write\_stream\_id**

Label0: Write Stream ID.

Type: `unsigned`

Default value: 0

**label10\_read\_ssd**

Label10: Read SDD or SSD\_Index.

Type: `unsigned`

Default value: 0

**label10\_read\_stream\_id**

Label10: Read Stream ID.

Type: unsigned

Default value: 0

**label10\_write\_ssd**

Label10: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label10\_write\_stream\_id**

Label10: Write Stream ID.

Type: unsigned

Default value: 0

**label11\_read\_ssd**

Label11: Read SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label11\_read\_stream\_id**

Label11: Read Stream ID.

Type: unsigned

Default value: 0

**label11\_write\_ssd**

Label11: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label11\_write\_stream\_id**

Label11: Write Stream ID.

Type: unsigned



Default value: 0

**label12\_read\_ssd**

Label12: Read SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label12\_read\_stream\_id**

Label12: Read Stream ID.

Type: unsigned

Default value: 0

**label12\_write\_ssd**

Label12: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label12\_write\_stream\_id**

Label12: Write Stream ID.

Type: unsigned

Default value: 0

**label13\_read\_ssd**

Label13: Read SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label13\_read\_stream\_id**

Label13: Read Stream ID.

Type: unsigned

Default value: 0

**label13\_write\_ssd**

Label13: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label13\_write\_stream\_id**

Label13: Write Stream ID.

Type: unsigned

Default value: 0

**label14\_read\_ssd**

Label14: Read SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label14\_read\_stream\_id**

Label14: Read Stream ID.

Type: unsigned

Default value: 0

**label14\_write\_ssd**

Label14: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label14\_write\_stream\_id**

Label14: Write Stream ID.

Type: unsigned

Default value: 0

**label15\_read\_ssd**

Label15: Read SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label15\_read\_stream\_id**

Label15: Read Stream ID.

Type: unsigned

Default value: 0

**label15\_write\_ssd**

Label15: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label15\_write\_stream\_id**

Label15: Write Stream ID.

Type: unsigned

Default value: 0

**label16\_read\_ssd**

Label16: Read SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label16\_read\_stream\_id**

Label16: Read Stream ID.

Type: unsigned

Default value: 0

**label16\_write\_ssd**

Label16: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label16\_write\_stream\_id**

Label16: Write Stream ID.

Type: unsigned

Default value: 0

**label17\_read\_ssd**

Label17: Read SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label17\_read\_stream\_id**

Label17: Read Stream ID.

Type: unsigned

Default value: 0

**label17\_write\_ssd**

Label17: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label17\_write\_stream\_id**

Label17: Write Stream ID.

Type: unsigned

Default value: 0

**label18\_read\_ssd**

Label18: Read SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label18\_read\_stream\_id**

Label18: Read Stream ID.

Type: unsigned

Default value: 0

**label18\_write\_ssd**

Label18: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label18\_write\_stream\_id**

Label18: Write Stream ID.

Type: unsigned

Default value: 0

**label19\_read\_ssd**

Label19: Read SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label19\_read\_stream\_id**

Label19: Read Stream ID.

Type: unsigned

Default value: 0

**label19\_write\_ssd**

Label19: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label19\_write\_stream\_id**

Label19: Write Stream ID.

Type: unsigned

Default value: 0

**label1\_read\_ssd**

Label1: Read SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label1\_read\_stream\_id**

Label1: Read Stream ID.

Type: unsigned

Default value: 0

**label1\_write\_ssd**

Label1: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label11\_write\_stream\_id**

Label11: Write Stream ID.

Type: unsigned

Default value: 0

**label20\_read\_ssd**

Label20: Read SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label20\_read\_stream\_id**

Label20: Read Stream ID.

Type: unsigned

Default value: 0

**label20\_write\_ssd**

Label20: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label20\_write\_stream\_id**

Label20: Write Stream ID.

Type: unsigned

Default value: 0

**label21\_read\_ssd**

Label21: Read SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label21\_read\_stream\_id**

Label21: Read Stream ID.

Type: unsigned

Default value: 0

**label21\_write\_ssd**

Label21: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label21\_write\_stream\_id**

Label21: Write Stream ID.

Type: unsigned

Default value: 0

**label22\_read\_ssd**

Label22: Read SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label22\_read\_stream\_id**

Label22: Read Stream ID.

Type: unsigned

Default value: 0

**label22\_write\_ssd**

Label22: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label22\_write\_stream\_id**

Label22: Write Stream ID.

Type: unsigned

Default value: 0

**label23\_read\_ssd**

Label23: Read SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label23\_read\_stream\_id**

Label23: Read Stream ID.

Type: unsigned

Default value: 0

**label23\_write\_ssd**

Label23: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label23\_write\_stream\_id**

Label23: Write Stream ID.

Type: unsigned

Default value: 0

**label24\_read\_ssd**

Label24: Read SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label24\_read\_stream\_id**

Label24: Read Stream ID.

Type: unsigned

Default value: 0

**label24\_write\_ssd**

Label24: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label24\_write\_stream\_id**

Label24: Write Stream ID.

Type: unsigned



Default value: 0

**label125\_read\_ssd**

Label25: Read SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label125\_read\_stream\_id**

Label25: Read Stream ID.

Type: unsigned

Default value: 0

**label125\_write\_ssd**

Label25: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label125\_write\_stream\_id**

Label25: Write Stream ID.

Type: unsigned

Default value: 0

**label126\_read\_ssd**

Label26: Read SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label126\_read\_stream\_id**

Label26: Read Stream ID.

Type: unsigned

Default value: 0

**label126\_write\_ssd**

Label26: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label126\_write\_stream\_id**

Label26: Write Stream ID.

Type: unsigned

Default value: 0

**label127\_read\_ssd**

Label27: Read SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label127\_read\_stream\_id**

Label27: Read Stream ID.

Type: unsigned

Default value: 0

**label127\_write\_ssd**

Label27: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label127\_write\_stream\_id**

Label27: Write Stream ID.

Type: unsigned

Default value: 0

**label128\_read\_ssd**

Label28: Read SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label128\_read\_stream\_id**

Label28: Read Stream ID.

Type: unsigned

Default value: 0

**label28\_write\_ssd**

Label28: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label28\_write\_stream\_id**

Label28: Write Stream ID.

Type: unsigned

Default value: 0

**label29\_read\_ssd**

Label29: Read SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label29\_read\_stream\_id**

Label29: Read Stream ID.

Type: unsigned

Default value: 0

**label29\_write\_ssd**

Label29: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label29\_write\_stream\_id**

Label29: Write Stream ID.

Type: unsigned

Default value: 0

**label2\_read\_ssd**

Label2: Read SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label12\_read\_stream\_id**

Label12: Read Stream ID.

Type: unsigned

Default value: 0

**label12\_write\_ssd**

Label12: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label12\_write\_stream\_id**

Label12: Write Stream ID.

Type: unsigned

Default value: 0

**label130\_read\_ssd**

Label130: Read SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label130\_read\_stream\_id**

Label130: Read Stream ID.

Type: unsigned

Default value: 0

**label130\_write\_ssd**

Label130: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label130\_write\_stream\_id**

Label130: Write Stream ID.

Type: unsigned

Default value: 0

**label31\_read\_ssd**

Label31: Read SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label31\_read\_stream\_id**

Label31: Read Stream ID.

Type: unsigned

Default value: 0

**label31\_write\_ssd**

Label31: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label31\_write\_stream\_id**

Label31: Write Stream ID.

Type: unsigned

Default value: 0

**label3\_read\_ssd**

Label3: Read SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label3\_read\_stream\_id**

Label3: Read Stream ID.

Type: unsigned

Default value: 0

**label3\_write\_ssd**

Label3: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label13\_write\_stream\_id**

Label13: Write Stream ID.

Type: unsigned

Default value: 0

**label14\_read\_ssd**

Label4: Read SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label14\_read\_stream\_id**

Label4: Read Stream ID.

Type: unsigned

Default value: 0

**label14\_write\_ssd**

Label4: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label14\_write\_stream\_id**

Label4: Write Stream ID.

Type: unsigned

Default value: 0

**label15\_read\_ssd**

Label5: Read SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label15\_read\_stream\_id**

Label5: Read Stream ID.

Type: unsigned

Default value: 0

**label5\_write\_ssd**

Label5: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label5\_write\_stream\_id**

Label5: Write Stream ID.

Type: unsigned

Default value: 0

**label6\_read\_ssd**

Label6: Read SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label6\_read\_stream\_id**

Label6: Read Stream ID.

Type: unsigned

Default value: 0

**label6\_write\_ssd**

Label6: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label6\_write\_stream\_id**

Label6: Write Stream ID.

Type: unsigned

Default value: 0

**label7\_read\_ssd**

Label7: Read SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label17\_read\_stream\_id**

Label17: Read Stream ID.

Type: unsigned

Default value: 0

**label17\_write\_ssd**

Label17: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label17\_write\_stream\_id**

Label17: Write Stream ID.

Type: unsigned

Default value: 0

**label18\_read\_ssd**

Label18: Read SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label18\_read\_stream\_id**

Label18: Read Stream ID.

Type: unsigned

Default value: 0

**label18\_write\_ssd**

Label18: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label18\_write\_stream\_id**

Label18: Write Stream ID.

Type: unsigned



Default value: 0

**label9\_read\_ssd**

Label9: Read SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label9\_read\_stream\_id**

Label9: Read Stream ID.

Type: unsigned

Default value: 0

**label9\_write\_ssd**

Label9: Write SDD or SSD\_Index.

Type: unsigned

Default value: 0

**label9\_write\_stream\_id**

Label9: Write Stream ID.

Type: unsigned

Default value: 0

**number\_of\_contexts**

Number of context banks.

Type: unsigned

Default value: 8

**number\_of\_smrs**

Number of stream match registers.

Type: unsigned

Default value: 32

**percent\_tlbstatus\_commits**

Percentage of times that a poll of TLBSTATUS will commit the TLBI commands.

Type: uint32\_t

Default value: 10

**prefetch\_only\_requests**

Handle prefetch-only requests by:- – deny them – use debug table walks and TLB entries – treat them as normal transactions (dangerous).

Type: unsigned

Default value: 0

**programmable\_non\_secure\_by\_default\_ssd\_indices**

Programmable SSD Indexes that are by default non-secure (e.g. 0, 6, 35-84).

Type: string

Default value: ""

**programmable\_secure\_by\_default\_ssd\_indices**

Programmable SSD Indexes that are by default secure (e.g. 0, 6, 35-84).

Type: string

Default value: ""

**ptw\_has\_separate\_port**

Page Table Walks use pvbus\_ptw\_m.

Type: bool

Default value: true

**supports\_nested\_translations**

Supports nested translations (stage 1 + stage 2).

Type: bool

Default value: true

**tlb\_depth**

TLB Depth (0 means 10000). The model will perform best with more TLB entries.

Type: unsigned

Default value: 2048

**use\_label\_mapping**

Use label mapping.

Type: `bool`

Default value: `true`

**use\_ssd\_determination\_table**

Use SSD Determination Table.

Type: `bool`

Default value: `true`

**version**

Version of the RTL that the model represents. Valid values are LACr1 and EAC.

Type: `string`

Default value: `"EAC"`

3.242 MMU\_500\_BASE

Defined in `LISA/SMMU_500_BASE.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r2p4	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

About SMMU\_500\_BASE

A SMMU that implements the architectural properties of an SMMU with particular specialization for the SMMU\_500.

NOTE that the SMMU-500 has to supply a 'StreamID' and potentially an 'SSD\_Index' to the underlying class

Iris and MTI instances for MMU\_500\_BASE

This model has the following Iris instances:

Name	Instance type
MMU_500_BASE	MMU_500_BASE
MMU_500_BASE.mapperY (where Y = 0-31)	PVBusMapper
MMU_500_BASE.ptw_dvm_receiver	PVBusMapper
MMU_500_BASE.ptw_master	PVBusMaster
MMU_500_BASE.pvbus_control_s_slv	PVBusSlave

Name	Instance type
MMU_500_BASE.pvbus_master	PVBusMaster
MMU_500_BASE.pvbusmasterY (where Y = 0-31)	PVBusMaster
MMU_500_BASE.pvbusslaveY (where Y = 0-31)	PVBusSlave

This model has the following MTI trace components:

Name	Component type
MMU_500_BASE.mapperY (where Y = 0-31)	PVBusMapper
MMU_500_BASE.ptw_dvm_receiver	PVBusMapper
MMU_500_BASE.ptw_master	PVBusMaster
MMU_500_BASE.pvbus_control_s_slv	PVBusSlave
MMU_500_BASE.pvbus_master	PVBusMaster
MMU_500_BASE.pvbusmasterY (where Y = 0-31)	PVBusMaster
MMU_500_BASE.pvbusslaveY (where Y = 0-31)	PVBusSlave

## Ports for MMU\_500\_BASE

Port	Direction	Protocol	Description
cfg_cttw_in	slave	Signal	The SoC supports coherent page walks, this is meant to be sampled at reset. However, in practice the model has to prevent the race condition between cfg_cttw being asserted at the same 'cycle' as negedge reset. Thus we actually only sample the signal on the first transaction to the SMMU or the first transition on this signal after reset. Thus in the model, we require that cfg_cttw be held for at least this period of time.
comb_irpt_ns	master	Signal	"Non-secure combined interrupt"
comb_irpt_s	master	Signal	"Secure combined interrupt"
cxt_irpt	master	Signal	Non-secure context bank fault.
glblflt_irpt_ns	master	Signal	Global non-secure Fault Interrupt In the SMMU Architecture this is called SMMU_NSglrpt.
glblflt_irpt_s	master	Signal	Global secure Fault Interrupt In the SMMU Architecture this is called SMMU_glirpt.
identify	master	MMU_500_BASE_IDENTIFY	This port is a special model port that is used to take a transaction and map it to an SSD/SSD_Index and StreamID.
priv_internals	slave	MMU_500_Internals	For internal use only, please do not use.
pvbus_control_s	slave	PVBus	The register port of the device is AXI.
pvbus_m	master	PVBus	This downstream port is where the translated accesses from pvbus_s emerge. See notes for pvbus_s[] as well. If the Page Table Walk (PTW) does not have a separate port then PTW accesses will emerge at port 0 with the same attributes as described in pvbus_ptw_m.

Port	Direction	Protocol	Description
pvbus_ptw_m	master	PVBus	This downstream port is where page table walk accesses come from. This is only used if configured to use a separate page table walk port. The MMU-500 will only obey DVM messages if configured to use this port. The page walks come out of this port with the following manager_id and user_flags. manager_id : 0xFFFFfff The user flags : user_flags[7:0] stage 1 context_id (or 0xFF if stage2 only) user_flags[15:8] stage 2 context_id (or 0xFF if stage 1 with stage 2 bypass) user_flags[18:16] stage 1 level user_flags[21:19] stage 2 level user_flags[31,30] adomain of the transaction NOTE that if the walk is being done for a stage 1 page walk descriptor fetch then the stage 1 level field will indicate that level. If the walk is being done for a stage 2 descriptor fetch, then the stage 2 level field will show that level. If the context-id for a stage is not valid (0xFF) then the 'level' information is 0x7.
pvbus_s	slave	PVBus	This port is the upstream port of the device, addresses on the port are in the VA/IPA Each TBU in the design is represented by a pair of pvbus_s[tbu_id] and pvbus_m[tbu_id]. That is transactions that go into pvbus_s[tbu_id] will emerge at pvbus_m[tbu_id]. The port index that a transaction comes in on is the tbu_number_ parameter to the MMU_500_BASE_IDENTIFY::identify() function. The identify() function must use all the information it is given by the parameters to map to the architectural concepts of StreamID and SSD_Index/SSD. How it does this is <b>IMPLEMENTATION DEFINED</b> and depends on the topology of the SoC and the masters upstream of the TBUs.
reset_in	slave	Signal	The reset pin.

## Parameters for MMU\_500\_BASE

### **PRIVATE\_PARAMETER\_personality**

The personality to use (affects ID codes and various imp def features).

Type: string

Default value: ""

### **PRIVATE\_PARAMETER\_seed**

Seed for randomised SMMU implementation defined behaviour.

Type: uint32\_t

Default value: 0x12345678

### **PRIVATE\_PARAMETER\_validation\_mode**

Internal validation mode.

Type: unsigned

Default value: 0

### **always\_secure\_ssd\_indices**

Non-programmable SSD Indexes that are always secure (e.g. 0, 6, 35-64).

Type: `string`

Default value: `""`

### **`cfg_cttw`**

Perform coherent page table walks.

Type: `bool`

Default value: `true`

### **`dump_unpredictability_in_user_flags`**

Override the user flags to encode unpredictable information (validation only).

Type: `bool`

Default value: `false`

### **`number_of_contexts`**

Number of context banks.

Type: `unsigned`

Default value: `8`

### **`number_of_smrs`**

Number of stream match registers.

Type: `unsigned`

Default value: `16`

### **`percent_tlbstatus_commits`**

Percentage of times that a poll of TLBSTATUS will commit the TLBI commands.

Type: `uint32_t`

Default value: `10`

### **`prefetch_only_requests`**

Handle prefetch-only requests by:- – deny them – use debug table walks and TLB entries – treat them as normal transactions (dangerous).

Type: `unsigned`

Default value: `0`

**programmable\_non\_secure\_by\_default\_ssd\_indices**

Programmable SSD Indexes that are by default non-secure (e.g. 0, 6, 35-84).

Type: `string`

Default value: `""`

**programmable\_secure\_by\_default\_ssd\_indices**

Programmable SSD Indexes that are by default secure (e.g. 0, 6, 35-84).

Type: `string`

Default value: `""`

**ptw\_has\_separate\_port**

Page Table Walks use `pvbus_ptw_m` (or uses `pvbus_m[0]`).

Type: `bool`

Default value: `true`

**supports\_nested\_translations**

Supports nested translations (stage 1 + stage 2).

Type: `bool`

Default value: `true`

**tlb\_depth**

TLB Depth (0 means 10000). The model will perform best with more TLB entries.

Type: `unsigned`

Default value: `2048`

**use\_ssd\_determination\_table**

Use SSD Determination Table.

Type: `bool`

Default value: `true`

**version**

Version of the RTL that the model represents. Valid values are LACr1 and EAC.

Type: `string`

Default value: `"EAC"`

## 3.243 MMU\_600

Defined in `LISA/SMMU_600.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support
r0p1	Full support
r0p2	Full support
r1p0	Full support
r2p0	Full support
r2p1	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Limitations

- No power control
- AMBA stash operations, destructive read and destructive hint operations are not supported on PVBUS and also are not supported by the device.
- The PMU has limited functionality. Only a subset of the architecturally mandatory events are supported, as indicated by the `SMMU_PMC_G_CBEID0` fields. The PMU is intended for demonstration purposes only and for driver development.



Between 11.17 and 11.18, the point of triggering for events changed for 'TLB miss' and this might lead to an (architecturally valid) change in the values captured in some circumstances.

- Limited RAS support
- The SYSCO interface is not implemented
- The low power interface is not implemented
- `TCU_CFG.XLATE_SLOTS` is fixed at 512
- `TCU_STATUS.GNT_XLATE_SLOTS` always reads at 512

### Notes

- A bad configuration renders the model inactive.
- Some configurations can be adjusted by configuration pins. These are only sampled at the negative edge of the reset pin. If you want to use these pins then you must drive them before sending a negative edge on the reset pin. During simulation reset the component driving them must also drive this transition again.
- Debug reads to the registers do not disturb the state.



- Writes to registers with Update flags, including debug writes, are ignored if the Update flag is already set to one.
- Debug and real accesses to the registers must be 32 bits or 64 bits.
- The model deals with groups of transactions with the same attributes and a similar range of addresses. The mapping used is remembered by the bus infrastructure and is used for subsequent sufficiently similar transactions without requiring intervention from the SMMU model, and is not traced, for instance.

The range of addresses that the mapping is valid for is determined by the SMMU model, depending on architectural and model-implementation details. However, as it is unaware of any sign-extension or the mapping that the SoC does, the SoC is responsible for subsequently narrowing the range of addresses for which this mapping is valid. Typically, this is done automatically when using PVBUSMapper.

- The hardware is a distributed SMMU and is divided into:
  - A single Translation Control Unit (TCU)
    - Has a port for the programming interface of the SMMU
    - Receives DVM messages
    - Does all the page walking, queue manipulation, etc.
  - One or more Translation Bus Units (TBUs)
    - Translate transactions from upstream (client) device into downstream transactions.
  - Zero or more connections to PCIe Root Complexes (PCIe-RCs)
  - There can be a total of 62 TBUs and PCIe-RCs attached.
  - An interconnect connecting the TBUs, PCIe-RCs to the TCU.
- A PCIe-RC has one connection to the TCU to make ATS requests but the PCIe-RC uses one or more TBUs to transform the transactions and pass them to the memory system. In the model, those TBUs are configured by the parameter `list_of_pcie_mode`. The SMMU does not know which TBUs a particular PCIe-RC is attached to.
- The TBU ORs a value into each StreamID that it receives. In the model, this is configured by the parameters:
  - `list_of_s_sid_high_at_bitpos0`
  - `list_of_ns_sid_high_at_bitpos0`
- The TCU, TBU, and the interconnect are all represented by this single model component.
- In the model, a pair of ports `tbs_pvbus_s[i]/tbm_pvbus_m[i]` represent a TBU 'i' or the `tbs_pvbus_s[i]` represents an incoming connection for a PCIe-RC. The corresponding reverse connection from the TCU to the PCIe-RC is by a special bus called `pvbus_id_routed_m` that is used to transport ATC Invalidates to the PCIe-RC.
  - In order to reduce system construction complexity the `tbs_pvbus_s[i]/tbm_pvbus_m[i]` also acts as a TBU so that the PCIe-RC need not separate its normal transactions and its ATS requests.
  - However, ATC Invalidates are only sent to a port which appears in `list_of_pcie_rc`. It should be uniquely decoded to a single port based on `list_of_ns_sid_high_at_bitpos0`

and those ATC Invalidates must be routed to the correct PCIe subsystem in order to invalidate the cache of ATS Response in the subsystem. Thus all TBUs that a PCIe-RC uses must have a unique reverse mapping from stream id to port.

- The pin `sup_oas` is not supported, instead it is a parameter as it is assumed that it would be tied to a fixed value in any specific platform.
- The hardware only has a single cacheability attribute for input transactions, but PVBUS transports both inner and outer cacheability.
- For non-PCIe-mode TBUs (i.e. their index does not appear in `list_of_pcie_mode` or `list_of_pcie_rc`):
  - For non-cache maintenance operations:
    - If the input attribute is any type of device then it is well defined as being outer-shared and Device-nGnRnE or Device-nGnRE. There is no support for Gathering or Reordering.
    - If the outer cacheable input attribute is normal then if it is Write-back then this is converted to Inner Write-back Outer Write-back (iWB-oWB) with the desired shareability. No Transient hint is supported and is always treated as non-transient.
    - This leaves all other normal memory types that are mapped to Inner Normal Non-cacheable, Outer Normal Non-cacheable outer shared (iNC-oNC-osh).
- Thus the upstream devices must present the cacheability in the *outer* cacheability attribute on PVBUS if it is cacheable. If it is a device type then both the inner and outer attributes must be set to the same. If it is iNC-oNC-osh then it must be presented as such.
- For PCIe-mode TBUs (i.e. their index appears in `list_of_pcie_mode` or `list_of_pcie_rc`):
  - Input transactions are from PCIe and the only indication of the memory type is in the NoSnoop bit of the transaction. No shareability is transported.
    - NoSnoop interpreted as iNC-oNC-osh
    - ! NoSnoop interpreted as iWB-oWB-ish (note inner shareable)
  - If a NoSnoop transaction has an attribute transform applied to it and the result of the transform is weaker than iNC-oNC-osh then it is forced to iNC-oNC-osh. For example, if a NoSnoop transaction uses a page table and is transformed to iWB-oWB-nsh then it is forced to iNC-oNC-osh. However, if the page tables transformed it to a device type then, as all device types are stronger than iNC-oNC-osh then it exits the SMMU as the device type.
  - In the model, transactions are classified by their incoming memory attributes as to whether they are NoSnoop or not and then normalized appropriately:
    - iWB-oWB-any-shareability transactions are interpreted as ! NoSnoop and therefore are normalized to iWB-oWB-ish.
    - Anything else is considered NoSnoop and therefore is normalized to iNC-oNC-osh.
  - Translated accesses also suffer the same interpretation to determine NoSnoop and how they are normalized. Thus they could enter the system with attributes different to if they were Untranslated Accesses translated by normal means.
  - It is expected that there is a component downstream of the SMMU that is aware of the system address map and will override the memory type to 'device' for any transaction that accesses a peripheral.

- The hardware has a single cacheability on input and, for transactions that are neither cache-maintenance operations nor PCIe transactions, normalizes the input to an architectural form before performing the SMMUv3 architectural transform:
  - Any device type is left untouched (the input can only represent Device-nGnRE and Device-nGnRnE).
  - If the input is Write-back (WB) then it is normalized to iWB-oWB with the incoming shareability.
  - If the input is anything else it is normalized to iNC-oNC-osh.
- The model accepts full architectural attributes of two levels of cacheability and so has to decide how to interpret this in terms of the hardware. For transactions that are not cache maintenance operations, the model replicates the outer attribute into the inner attribute and then performs the normalization that the hardware does.
- The hardware normalizes the architectural output attributes and outputs a single level of cacheability and a user flag (OC) specifying if the architectural attributes were cacheable in the outer cacheable domain. If the transaction is classified as a PCIe one then the NoSnoop transform described above is applied. If the original transaction was NoSnoop, then any weaker memory type is strengthened to iNC-oNC-osh so apply the following transform:

```

if      iWB-oWB-nsh/ish/osh      then output WB-nsh/ish/osh, OC = 1
else if i (NC/WT/WB) -o (WB/WT)  then output NC-Sys,      OC = 1
else if i (NC/WT/WB) -oNC        then output NC-Sys,      OC = 0
else if Device-(GRE/nGRE/nGnRE)  then output DV-Sys,      OC = 0
else                             output SO-Sys,          OC = 0

```

- The model only normalizes according to PCIe but otherwise leaves the architectural attributes intact on the output bus.
- MSIs are issued on the `qtw_pvbuss_m` port using attributes determined by the parameter `msi_attribute_transform`, whilst Event Queue writes are always issued with `ExtendedID=0`, `UserFlags=0`, `ManagerID64=0xFFFFfff`.

In the hardware, there is no way of distinguishing Event Queue writes from MSI writes, however, this provides a mechanism that if the model system needs to distinguish then it can.

## Parameters for parsing transaction attributes

Some of the parameters are strings that share a common parsing format for extracting from and placing information into the transaction's attributes.

They can extract and place information into the following fields of a transaction's attributes:

- `ManagerID64` (64 bits)
- `ExtendedID` (64 bits)
- `UserFlags` (32 bits)

The string parameter is parsed as a comma-separated list of:

```
lhs_expression=rhs_expression
```

The `lhs_expression` and `rhs_expression` can be an entire symbol, for example:

```
StreamID
```

or a bit, or bit slice:

```
StreamID[16]  
StreamID[31:20]
```

In `rhs_expression`, numeric constants (or slices of numeric constants) are also allowed:

```
StreamID[16]=1  
StreamID[16]=10[2]
```

A single symbol might be assigned from multiple non-contiguous arrays of bits from a mix of different RHS symbols:

```
StreamID[16]=1, StreamID[31:30]=ExtendedID[1:0],  
StreamID[15:0]=UserFlags[31:16], ...
```

In those cases where a left hand symbol can also appear on the right hand side, it is possible to swap bits and transform the symbol. For example, the following expression swaps bits 0 and 1 of the `ExtendedID`:

```
ExtendedID[0]=ExtendedID[1], ExtendedID[1]=ExtendedID[0]
```

Any bits of the attributes that have no transform specified are retained from the input.

The `lhs_expression` and `rhs_expression` must have the same bit width.

**Iris and MTI instances for MMU\_600**

This model has the following Iris instances:

Name	Instance type
MMU_600	MMU_600
MMU_600.axi_stream_msi_manager[4294967295].pvbusmaster	PVBusMaster
MMU_600.pvbus_id_routed_m[Y].pvbusmaster (where Y = 0,4294967295)	PVBusMaster
MMU_600.register_file[0]	PVBusSlave
MMU_600.service_request_tbu[Y] (where Y = 0-63)	PVBusSlave
MMU_600.tbu[Y] (where Y = 0-63)	PVBusMapper
MMU_600.tw_msi_qs_manager[4294967295].pvbusmaster	PVBusMaster

This model has the following MTI trace components:

Name	Component type
MMU_600	MMU_600
MMU_600.axi_stream_msi_manager[4294967295].pvbusmaster	PVBusMaster
MMU_600.pvbus_id_routed_m[Y].pvbusmaster (where Y = 0,4294967295)	PVBusMaster
MMU_600.register_file[0]	PVBusSlave
MMU_600.service_request_tbu[Y] (where Y = 0-63)	PVBusSlave
MMU_600.tbu[Y] (where Y = 0-63)	PVBusMapper
MMU_600.tw_msi_qs_manager[4294967295].pvbusmaster	PVBusMaster

### Ports for MMU\_600

Port	Direction	Protocol	Description
clk_in	slave	ClockSignal	Clock signal (in RTL aclk) This is a clock time-base used by the TCU to spread some of its processing over time, if enabled by the wait_* parameters. The clock must always be connected.
cmd_sync_irpt_ns	master	Signal	Pulsed interrupt output signal for non-secure CMD_SYNC having a completion signal of SIG_IRQ.
cmd_sync_irpt_s	master	Signal	Pulsed interrupt output signal for secure CMD_SYNC having a completion signal of SIG_IRQ.
event_q_irpt_ns	master	Signal	Pulsed interrupt output signal for the non-secure event queue becoming non-empty.
event_q_irpt_s	master	Signal	Pulsed interrupt output signal for the secure event queue becoming non-empty.
evento	master	Signal	Event signal
global_irpt_ns	master	Signal	Pulsed interrupt output signal for non-secure SMMU_GERROR(N) signalling an error.
global_irpt_s	master	Signal	Pulsed interrupt output signal for secure SMMU_S_GERROR(N) signalling an error.
identify	master	SMMUv3AEMIdentifyProtocol	Map the transaction to the tuple (StreamID, SubStreamID, SubStreamIDValid, SSD) The StreamID that is produced by the implementation of this protocol is not the final StreamID. The final StreamID is produced by using the list_of_ns_sid_high_at_bitpos0/ list_of_s_sid_high_at_bitpos0 parameter to map the StreamID based on the upstream port index. Also see the parameter howto_identify which can replace the functionality of this port under certain circumstances.
pri_q_irpt_ns	master	Signal	Pulsed interrupt output signal for the PRI queue becoming non-empty. Exists only for r1 and higher.
prog_pvbus_s	slave	PVBus	Register subordinate port (in RTL PROG)
pvbus_id_routed_m	master	PVBus	This is a special "id-routed" port for transmitting ATC invalidates upstream into the PCIe EndPoints, it is not a normal bus. See the parameter output_id_routed_transform. The FastModels ATC Invalidate and PRI Response protocol specifies how to route and deal with this port.
qtw_pvbus_m	master	PVBus	Manager port used for Table Walks, MSIs and Queue access. Note that although this is a manager port, it can still receive DVM messages.

Port	Direction	Protocol	Description
sec_override	slave	Signal	Allow certain registers to be accessible to non-secure accesses from reset, as described in the TCU_SCR register. This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.
sup_btm	slave	Signal	System supports BTM and will be reflected in the IDR registers. This signal can override the value set by the parameters configuring the IDR registers. If BTM (Broadcast Table Maintenance) is not supported then DVM messages will be ignored. This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.
sup_cohacc	slave	Signal	System supports COHACC and will be reflected in the IDR registers. This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.
sup_sev	slave	Signal	System supports SEV and will be reflected in the IDR registers. This signal can override the value set by the parameters configuring the IDR registers. This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.
tbm_pvbus_m	master	PVBus	The TBU manager ports that carry transactions that have been translated from the correspondingly numbered tbs_pvbus_s[] port.
tbs_pvbus_s	slave	PVBus	The TBU subordinate ports that receive transactions to be translated. They will exit the SMMU through the same numbered pvbus_m[] port.
tbu_pmu_irpt	master	Signal	TBU Performance Monitoring Unit interrupt, one per TBU.
tbu_pmusnapshot_ack	master	Signal	PMU snapshot interface for the TBU, ack a snapshot.
tbu_pmusnapshot_req	slave	Signal	PMU snapshot interface for the TBU, request a snapshot.
tbu_ras_irpt	master	Signal	The RAS interrupt pin for errors detected in the TBUs.
tbu_reset_in	slave	Signal	Reset signals The TBUs can have independent reset signals. However, TCU reset will result in the reset of all TBUs. This behavior is unlike SMMU RTL implementations which do allow for independent reset of the TCU separate from TBUs. Each signal tbu_reset_in[n] corresponds to the TBU using tbs_pvbus_s[n]/tbs_pvbus_s[n] pair. If the SMMU receives a transaction whilst the TBU is expected to be in reset then it will complain using the ArchMsg.Warning.warning trace source. Those tbu_reset_in that correspond to a PCIe-RC connection can be connected to monitor the PCIe-RC's reset signal. If it receives an ATS request when in reset then it will complain in a similar way. You must connect these pins if you wish the TCU_NODE_STATUS for the nodes to be accurate (including any connected to the PCIe-RC).
tcu_pmu_irpt	master	Signal	TCU Performance Monitoring Unit interrupt
tcu_pmusnapshot_ack	master	Signal	PMU snapshot interface for the TCU, ack a snapshot.
tcu_pmusnapshot_req	slave	Signal	PMU snapshot interface This is a four-phase handshake to have the corresponding PMCG perform a capture of the current counter values. PMU snapshot interface for the TCU, request a snapshot.
tcu_ras_irpt	master	Signal	The RAS interrupt pin for errors detected in the TCU.
tcu_reset_in	slave	Signal	The reset signal to the TCU interface.

## Parameters for MMU\_600

### TCUCFG\_XLATE\_SLOTS

Maximum number of outstanding stalled transactions that the SMMU supports.



TCUCFG\_XLATE\_SLOTS must be  $\geq$  TCUCFG\_PTW\_SLOTS which is currently fixed to 512.

---

Accepted Values: 512.

Type: `uint32_t`

Default value: 512

### all\_error\_messages\_through\_trace

Some conditions in the SMMU are so strange that the software programming the SMMU has done something wrong. At this point messages are output to either `ArchMsg.Error.*` or `ArchMsg.Warning.*` or to the error stream of the simulator. Outputting to the error stream of the simulator may cause it to return with a non-zero exit status.

If you set this option to true then instead of using the error stream of the simulator it will always use a trace stream allowing the simulation to exit with a zero exit status.

Type: `bool`

Default value: `false`

### behaviour\_of\_sampled\_at\_reset\_signals

Some configuration signals into the SMMU are sampled on negedge of reset.

However, it can sometimes be hard to arrange to drive a configuration pin before the negedge of reset.

The configuration pins are sampled:

**0**

at negedge reset.

**1**

at negedge reset, but if a later change occurs at the same simulated time, and no transactions have occurred, then they will be resampled and the SMMU reset again.

Type: `unsigned`

Default value: 0

**cmdq\_max\_number\_of\_commands\_to\_buffer**

The command queues can buffer fetched commands before issuing them. This parameter is roughly the maximum number of commands to do this for. The programmer visible effects are that just because the CONS pointer shows a command has been consumed does not necessarily mean that it has been issued (and completed). Higher values accentuate this effect.

Type: `uint32_t`

Default value: 10

**enable\_device\_id\_checks**

If this parameter is true then the DeviceIDs seen by the GIC are:

- **for client devices**

`DeviceID = StreamID + translated_device_id_base`

- **for SMMU-generated MSIs**

`smmu_msi_device_id`

This parameter enables two checks:

- If the DeviceID is used in the `output_attribute_transform` parameter, if it overflows 32 bits then the model warns. If the DeviceID is not used, it is assumed that the external agent that forms the DeviceID warns if it overflows.
- If the SMMU supports MSIs, the model checks that the GIC is able to distinguish an MSI generated by the SMMU from one generated by a client device.

As the exact mechanism to determine the DeviceID is in the system and not necessarily under control of the SMMU then you can disable these warnings using this parameter.

See also the parameters: `output_attribute_transform` and `msi_attribute_transform`.

Type: `bool`

Default value: `true`

**howto\_identify**

If `use-identify` then the SMMU uses the `identify` port to determine the `ssd`, `StreamID`, `SubStreamID`. Otherwise, this string extracts them from the transaction's attributes.

Examples:

```
SEC_SID=ExtendedID[63], SSV=ExtendedID[62], SubstreamID=ExtendedID[51:32],
StreamID=ExtendedID[31:0]
```

or

```
nSEC_SID=ExtendedID[63], StreamID=ExtendedID[55:24], nSSV=ExtendedID[20],
SubstreamID=ExtendedID[19:0]
```



```
StreamID[31:24]=0, StreamID[23:0]=ExtendedID[23:0], SSV=1[0], ...
```

**Note**

If you are not using the `use-identify` option then the configuration string is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

LHS Symbols:

**SIDV**

Indicates that the StreamID is valid.

**SSV**

Indicates that the SubstreamID is valid.

**SEC\_SID / SEC\_SID\_bit\_1**

Bits 0 and 1 respectively of the StreamID Security State

**StreamID**

(32 b) valid if `SIDV` is 1 or both `SIDV` and `nSIDV` are unused.

**SubstreamID**

(20 b) is valid if `ssv` is true.

**nSEC\_SID / nSEC\_SID\_bit\_1 / nSSV / nSIDV**

Negative logic of above parameters. Different attributes are independent and can use negative or positive logic.

RHS Symbols:

**ExtendedID / ManagerID64 / UserFlags**

Incoming PVBUS transaction attributes

`SEC_SID` is 1b in the model. For RME systems, in hardware, then `SEC_SID` is 2b, in the model `SEC_SID_bit_1` represents bit[1].

`SIDV == 0` indicates a NoStreamID transaction and the SSD is the PAS of the transaction, `SEC_SID` is not used.

Negative and positive logic symbols for the same attribute is an error.

The model uses the term SSD (Security State Determination) to mean which security state the transaction, register, structure, event, etc. belongs to.

In the SMMUV3 Architecture, the side-band signal `SEC_SID` holds the information for the transactions, but uses a different encoding.

Before RME, `SEC_SID` was a one bit signal. With RME, in the hardware, it was expanded to 2b. To retain backwards compatibility in the model, `SEC_SID` remains as 1b in this parameter, but the second bit can be expressed with `SEC_SID_bit_1` (and its negative logic version `nSEC_SID_bit_1`)

Security state	SSD	SEC_SID_bit_1	SEC_SID
secure	0	0	1
non-secure	1	0	0
root (reserved)	2	1	1
realm	3	1	0

For those systems that support NoStreamID transactions, and `howto_identify` is not using the port, first the SMMU determines the value of `sidv` or `nsidv` to see if the transaction is a NoStreamID transaction (`sidv == 0` or `nsidv == 1`). If so then the rest of the `howto_identify` string is ignored as the information extracted relates only to StreamID transactions. Instead more information is extracted using the parameter `howto_identify_NoStreamID_extra_info`.

In AMBA systems, the equivalent signal to `sidv` is called either `ARMMUVALID` or `AWMMUVALID`. Collectively they are known as `AxMMUVALID`.

Type: `string`

Default value: `"use-identify"`

### **`list_of_ns_sid_high_at_bitpos0`**

A comma-separated list of values to bitwise OR into each Non-secure StreamID for each TBU/Node. Bit 0 of the value corresponds to bit 0 of the StreamID.

Each TBU that is connected to a PCIe-RC (see `list_of_pcie_rc`) must serve a unique contiguous subset of StreamIDs as determined by their top bits. This is used in order to know which port to route ATC Invalidates and PRI Responses to the PCIe subsystems.

The empty string corresponds to all 0s.

Type: `string`

Default value: `""`

### **`list_of_pcie_mode`**

A comma-separated list of ranges of ports that represent TBUs that are attached to PCIe Root Complexes (PCIe-RC). A single PCIe-RC might use several TBUs and stripe accesses across them. The attribute handling for these TBUs is slightly different in that if the PCIe transaction is NoSnoop and the output attributes of the translation would be weaker than `inc-onc-osh` then the output is forced to `inc-onc-osh`.

`inc-onc-osh == "inner normal non-cacheable, out normal non-cacheable, outer shared"`.

Type: `string`

Default value: `""`

**list\_of\_pcie\_rc**

This is a list of ports that are connected to PCIe Root Complex (PCIe-RC) by a protocol called DTI-ATS. This port is used to transport ATS and PRI Requests to the SMMU from the PCIe-RC.

In the real hardware, the PCIe-RC uses this port for ATS/PRI, and the actual transactions go through separate TBUs. In the model, this port can accept actual transactions as well. However, in the model, then the ATC Invalidates and the PRI Responses need to be transferred over the corresponding pvbush\_id\_routed\_m port as DTI-ATS is bidirectional, but PVBush is not.

Type: `string`

Default value: `""`

**list\_of\_s\_sid\_high\_at\_bitpos0**

A comma-separated list of values to bitwise OR into each Secure StreamID for each TBU/Node. Bit 0 of the value corresponds to bit 0 of the StreamID.

The empty string corresponds to all 0s.

`use-ns` can be used to apply the `list_of_ns_sid_high_at_bitpos0` values instead.

Type: `string`

Default value: `""`

**msi\_attribute\_transform**

Transform downstream attributes of MSI transactions.



The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

---

- `""` or `"none"` – no transform
- How to alter output attributes of SMMU-generated MSIs. Example:

```
"UserFlags[15:0]=smmu_msi_device_id[31:16],
ManagerID64[15:0]=smmu_msi_device_id[15:0],
ExtendedID=0"
```

LHS Symbols:

**ExtendedID / ManagerID64 / UserFlags**

Outgoing PVBush transaction attributes

RHS Symbols:

**smmu\_msi\_device\_id**

The value stored in the parameter with the same name

**interrupt\_kind**

The selected bit corresponds to the interrupts listed below

**0/1**

EVENTQ s/ns

**2**

PRIQ

**3/4**

CMD\_SYNC s/ns

**5/6**

GERROR s/ns

**7/8**

PMCG s/ns

**9/10/11**

RAS FHI/ERI/CRI

**12/13**

gpf\_far/gpt\_cfg\_far

**14/15/16/17**

Realm EVENTQ/PRIQ/CMDQ/GERROR

**HWATTR\_KIND\_0**

PBHA information

**Numeric Literals**

Any number. Ex: 0x1234

ExtendedID/ManagerID64/UserFlags start with values {0, 0xFFFFFFFF, 0} respectively.

Any bits with no transform are unchanged.

This transform can be used to determine the DeviceID passed to the GIC to distinguish MSIs generated by the SMMU from those generated by client devices.



**Note**

See also `output_attribute_transform` and `enable_device_id_checks`.

---



After 11.25 the `interrupt_kind` field was extended to 5 bits. This is strictly a non-backwards compatible change. However, the original 3 bits were insufficient to express all the interrupt kinds that exist.

Type: `string`

Default value: `"ExtendedID[31:0]=smmu_msi_device_id, ManagerID64[31:0]=0xFFFFFFFF"`

### **number\_of\_ports**

The number of port pairs that the SMMU has.

Type: `unsigned`

Default value: 1

### **output\_attribute\_transform**

Transform the downstream attributes of a translated transaction.



The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

- "" or "none": no transform
- How to alter output attributes. Example:

```
"ExtendedID[15:0]=DeviceID[15:0], UserFlags[31]=nSSV,
UserFlags[19:0]=SubstreamID,
ManagerID64[10]=ManagerID64[11], ManagerID64[11]=ManagerID64[10]"
```

RHS/LHS Symbols:

#### **ExtendedID / ManagerID64 / UserFlags**

Incoming/Outgoing PVBUS transaction attributes

RHS Symbols:

#### **DeviceID**

`StreamID + translated_device_id_base`

#### **StreamID / SubstreamID / SEC\_SID / SSV**

Architectural information. See parameter `howto_identify` for more information.

#### **nSEC\_SID / nSSV**

Negative logic of above parameters. Different attributes are independent and can use negative or positive logic.

**St1PBHA / St2PBHA**

Page Based Hardware Attributes from leaf descriptors (zero if unused).

**STE\_IMPDEF1**

STE[127:116]

**Numeric Literals**

Any number. Ex: 0x1234

The streamID has had ns\_sid\_high/s\_sid\_high ORred into it for the appropriate TBU.

Type: string

Default value: "ExtendedID[31:0]=DeviceID"

**output\_id\_routed\_transform**

The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

The SMMU generates the following ID-routed transaction on the pvbus\_id\_routed\_m bus:

- ATC Invalidate
- PRI Response

This parameter controls how the SMMU should express:

- the StreamID
- the Trusted (T) bit

The value is a comma-separated list of assignments:

```
Address[27:12]=StreamID[15:0], ExtendedID[60]=T, ExtendedID[15:0]=StreamID[31:16]
```

Address bits[11:0] cannot be used.

The LHS can be one of:

- PAS
- ManagerID64 / ExtendedID / UserFlags
- Address

The RHS can be one of:

- a numeric constant
- SSD

- $T$  or negative version  $nT$
- `StreamID`

For realm (or 'Trusted') transactions, then  $ssd=0b11$ ,  $T=1$ ,  $nT=0$ . For non-secure (or 'Non-Trusted') transactions, then  $ssd=0b01$ ,  $T=0$ ,  $nT=1$ .

Type: `string`

Default value: `"Address[27:12]=StreamID[15:0], PAS=SSD"`

### **prefetch\_only\_requests**

The simulator supports 'prefetch-only' DMI requests, which can occur at any time and for any reason and are intended to be invisible to the end execution of the model and to the user.

**0**

deny all prefetch-only requests

**1**

- use debug requests for any page table walks
  - form and use debug TLB/cache entries
  - any faults will not record, but deny the prefetch request

**2**

- treat prefetch-only requests like normal transactions
  - use normal page table walk transactions
  - use and form normal TLB/cache entries
  - faults will alter the programmer-visible state of the SMMU

0 is the safest.

1 treats the access like a debug request and requires that debug page table walks are treated correctly downstream. Any descriptors that need HTTU to allow the transaction to proceed will fail the request.

2 is dangerous, it uses real transactions and reports faults that are unphysical. Real transactions can be `wait()`ed and this disobeys the SystemC spec for `get_direct_mem_ptr()`.

Type: `unsigned`

Default value: 0

### **sec\_override**

The IMP DEF port `sec_override` controls whether some of the registers are accessible to secure or non-secure transactions. This parameter is the default value assumed for that port if the port is not driven by a signal.

Type: `bool`

Default value: `false`

### **seed**

Used to seed the pseudo-random number generator that the SMMU model uses.

Type: `uint32_t`

Default value: `0x12345678`

### **size\_of\_cd\_cache**

The number of entries in the cache holding CD structures. If this is zero it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

Type: `uint32_t`

Default value: `0`

### **size\_of\_llcd\_cache**

The number of entries in the cache holding L1CD descriptors. If this is zero it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

Type: `uint32_t`

Default value: `0`

### **size\_of\_llste\_cache**

The number of entries in the cache holding L1STE descriptors. If this is zero it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

Type: `uint32_t`

Default value: `0`

### **size\_of\_ste\_cache**

The number of entries in the cache holding STE structures. If this is zero it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

Type: `uint32_t`

Default value: `0`

### **size\_of\_tlb**

The number of entries in the TLB. If this is zero it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

Type: `uint32_t`

Default value: `0`



**smmu\_msi\_device\_id**

When appropriately enabled, assume that MSIs that are generated by the SMMU are presented to the GIC with this DeviceID.

See parameters `msi_attribute_transform` and `enable_device_id_checks`.

Type: `uint32_t`

Default value: `0`

**sup\_btm**

The default value of the register field `SMMU_IDR0.BTM` and `SMMU_ROOT_IDR0.BGPTM` (when supported). This enables Broadcast TLB maintenance.

Type: `bool`

Default value: `true`

**sup\_cohacc**

The default value of the register `SMMU_IDR0.COACC`.

Type: `bool`

Default value: `true`

**sup\_oas**

The hardware has an input port `sup_oas[2:0]` that indicates what output address size (OAS) the system has. This is sampled at reset.

The model does not have this port as it is expected to be a constant for the system and not to change. Instead it is just a parameter.

The allowed values are:

- 0**  
32 bits
- 1**  
36 bits
- 2**  
40 bits
- 3**  
42 bits
- 4**  
44 bits

**5**

48 bits.

Type: unsigned

Default value: 5

**sup\_sev**The default value of the register `SMMU_IDR0.SEV`.

Type: bool

Default value: true

**tlb\_when\_do\_f\_tlb\_conflict\_on\_overlap**

If a TLB entry is created by a walk and it overlaps an existing entry, there are some architectural situations where the result is known. For all others, then an implementation is allowed to use an **UNPREDICTABLE** combination of the two entries, or it can generate `F_TLB_CONFLICT`:

**0**

never generate

**1**

sometimes generate

**2**

always generate

Conflicts between global and non-global entries are not detected by the model.

Type: unsigned

Default value: 0

**translated\_device\_id\_base**

When appropriately enabled, assume that client device accesses are translated to a DeviceID as seen by the GIC of:

```
StreamID + translated_device_id_base
```

See parameter `output_attribute_transform` and `enable_device_id_checks`.

Type: uint32\_t

Default value: 0

**tw\_qs\_attribute\_transform**

Transform downstream attributes of table walk and queue transactions.



The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

- "" or "none" – no transform
- How to alter the output attributes. Example:

```
"ExtendedID[35:32]=HWATTR_KIND_0"
```

RHS/LHS Symbols:

**ExtendedID / ManagerID64 / UserFlags**

Incoming/Outgoing PVBUS transaction attributes

RHS Symbols:

**HWATTR\_KIND\_0**

PBHA information

**kind**

Transaction kind

- For a read:

**0/1**

L1STE/STE

**2/3**

L1CD/CD

**4/5**

S1/S2 TTD (including CAS)

**6**

CMDQ

**7**

VMS

**11/12**

LOGPT/L1GPT

**13/14**

LODPT/L1DPT

- For a write:

**0**

EVENTQ

**1**

PRIQ

ExtendedID/ManagerID64/UserFlags start with values {0, 0xFFFFFFFF, 0} respectively.

Any bits with no transform are unchanged.



See also `output_attribute_transform` and `msi_attribute_transform`.

---

Type: string

Default value: ""

### **version**

The version of this product.

Type: string

Default value: "r0p0"

### **wait\_cmdq\_ticks**

This is the time to wait before doing something on the command queue. If bit [32] is set 0x1\_0000\_0000 then the time waited for is a uniform randomly-distributed time  $[0, \max(2, (t \& 0xFFFFFFFF)) - 1]$ .

Type: uint64\_t

Default value: 0

### **wait\_eventq\_ticks**

This is the time to wait before doing something on the event queue. If bit [32] is set 0x1\_0000\_0000 then the time waited for is a uniform randomly-distributed time  $[0, \max(2, (t \& 0xFFFFFFFF)) - 1]$ .

Type: uint64\_t

Default value: 0

### **wait\_imp\_def\_work\_ticks**

This is the time to wait before doing an IMP DEF operation. If bit 32 is set (0x1\_0000\_0000) then the time waited for is a uniform randomly distributed time  $[0, \max(2, (t \& 0xFFFFFFFF)) - 1]$ .

The IMP DEF work in this case is the number of ticks between raising pmusnapshot\_req and pmusnapshot\_ack being raised, and the converse operation.

Type: uint64\_t

Default value: 0

**wait\_misc\_async\_actions\_ticks**

This is the time to wait before doing an async action that could be delayed is performed. If bit [32] is set 0x1\_0000\_0000 then the time waited for is a uniform randomly-distributed time  $[0, \max(2, (t \& 0xFFFFFFFF)) - 1]$ .

Type: uint64\_t

Default value: 0

**wait\_msi\_ticks**

This is the time to wait before sending an MSI. If bit [32] is set 0x1\_0000\_0000 then the time waited for is a uniform randomly-distributed time  $[0, \max(2, (t \& 0xFFFFFFFF)) - 1]$ .

Type: uint64\_t

Default value: 0

**wait\_pri\_req\_ticks**

This is the time to wait before processing a PRI Request. If bit [32] is set 0x1\_0000\_0000 then the time waited for is a uniform randomly-distributed time  $[0, \max(2, (t \& 0xFFFFFFFF)) - 1]$ .

Type: uint64\_t

Default value: 0

**wait\_pri\_resp\_ticks**

This is the time to wait before sending a PRI Response back to the PCIe subsystem. When a PRI Response is an auto-response then the ATC might immediately make a new ATS request, that immediately fails, immediately makes a PRI Request, or auto-responds, etc. To break this loop, then we introduce a minimum time on all PRI Responses to give other components in the system a chance to run. If bit [32] is set 0x1\_0000\_0000 then the time waited for is a uniform randomly-distributed time  $[0, \max(2, (t \& 0xFFFFFFFF)) - 1]$ .

Type: uint64\_t

Default value: 1

3.244 MMU\_700

Defined in LISA/SMMU\_700.lisa.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support
r1p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

## Limitations

- No power control
- AMBA stash operations, destructive read and destructive hint operations are not supported on PVBUS and also are not supported by the device.
- The PMU has limited functionality. Only a subset of the architecturally mandatory events are supported, as indicated by the `SMMU_PMCG_CEID0` fields. The PMU is intended for demonstration purposes only and for driver development.



Between 11.17 and 11.18, the point of triggering for events changed for 'TLB miss' and this might lead to an (architecturally valid) change in the values captured in some circumstances.

---

- Limited RAS support
- The SYSCO interface is not implemented
- The low power interface is not implemented
- The IMP DEF MPAM register file is not implemented. This controls how the internal resources of the MMU-700 are partitioned.
- `tcu_sid[31:0]` is not modelled, instead the parameter `smmu_msi_device_id` is used.
- The HWATTR side-band signal is available by configuring `output_attribute_transform`, `msi_attribute_transform` and `tw_qs_attribute_transform` with `HWATTR_KIND_0`. For some transactions HWATTR comes from the `SMMU_S_AGBPA[3:0]`. In the hardware, this register has an 'Update' bit[31] that should be written as 1 and will be turned to zero when all transactions using the old value have completed. The model does not implement this behavior and the Update bit is **RAZ/WI**.
- Any configuration parameter listed in the TRM but not shown in this file is not supported.
- `TCU_STATUS.GNT_XLATE_SLOTS` always reads at 512

## Notes

- A bad configuration renders the model inactive.
- Some configurations can be adjusted by configuration pins. These are only sampled at the negative edge of the reset pin. If you want to use these pins then you must drive them before sending a negative edge on the reset pin. During simulation reset the component driving them must also drive this transition again.
- Debug reads to the registers do not disturb the state.
- Writes to registers with Update flags, including debug writes, are ignored if the Update flag is already set to one.
- Debug and real accesses to the registers must be 32 bits or 64 bits.
- The model deals with groups of transactions with the same attributes and a similar range of addresses. The mapping used is remembered by the bus infrastructure and is used for

subsequent sufficiently similar transactions without requiring intervention from the SMMU model, and is not traced, for instance.

The range of addresses that the mapping is valid for is determined by the SMMU model, depending on architectural and model-implementation details. However, as it is unaware of any sign-extension or the mapping that the SoC does, the SoC is responsible for subsequently narrowing the range of addresses for which this mapping is valid. Typically, this is done automatically when using PVBUSMapper.

- The hardware is a distributed SMMU and is divided into:
  - A single Translation Control Unit (TCU)
    - Has a port for the programming interface of the SMMU
    - Receives DVM messages
    - Does all the page walking, queue manipulation, etc.
  - One or more Translation Bus Units (TBUs)
    - Translate transactions from upstream (client) device into downstream transactions.
  - Zero or more connections to PCIe Root Complexes (PCIe-RCs)
  - There can be a total of 62 TBUs and PCIe-RCs attached.
  - An interconnect connecting the TBUs, PCIe-RCs to the TCU.
- A PCIe-RC has one connection to the TCU to make ATS requests but the PCIe-RC uses one or more TBUs to transform the transactions and pass them to the memory system. In the model, those TBUs are configured by the parameter `list_of_pcie_mode`. The SMMU does not know which TBUs a particular PCIe-RC is attached to.
- The TBU ORs a value into each StreamID that it receives. In the model, this is configured by the parameters:
  - `list_of_s_sid_high_at_bitpos0`
  - `list_of_ns_sid_high_at_bitpos0`
- The TCU, TBU, and the interconnect are all represented by this single model component.
- In the model, a pair of ports `tbs_pvbuss[i]/tbm_pvbuss_m[i]` represent a TBU 'i' or the `tbs_pvbuss[i]` represents an incoming connection for a PCIe-RC. The corresponding reverse connection from the TCU to the PCIe-RC is by a special bus called `pvbus_id_routed_m` that is used to transport ATC Invalidates to the PCIe-RC.
  - In order to reduce system construction complexity the `tbs_pvbuss[i]/tbm_pvbuss_m[i]` also acts as a TBU so that the PCIe-RC need not separate its normal transactions and its ATS requests.
  - However, ATC Invalidates are only sent to a port which appears in `list_of_pcie_rc`. It should be uniquely decoded to a single port based on `list_of_ns_sid_high_at_bitpos0` and those ATC Invalidates must be routed to the correct PCIe subsystem in order to invalidate the cache of ATS Response in the subsystem. Thus all TBUs that a PCIe-RC uses must have a unique reverse mapping from stream id to port.
- The pin `sup_oas` is not supported, instead it is a parameter as it is assumed that it would be tied to a fixed value in any specific platform.

- The hardware only has a single cacheability attribute for input transactions, but PVBUS transports both inner and outer cacheability.
- For non-PCIe-mode TBUs (i.e. their index does not appear in `list_of_pcie_mode` or `list_of_pcie_rc`):
  - For non-cache maintenance operations:
    - If the input attribute is any type of device then it is well defined as being outer-shared and Device-nGnRnE or Device-nGnRE. There is no support for Gathering or Reordering.
    - If the outer cacheable input attribute is normal then if it is Write-back then this is converted to Inner Write-back Outer Write-back (iWB-oWB) with the desired shareability. No Transient hint is supported and is always treated as non-transient.
    - This leaves all other normal memory types that are mapped to Inner Normal Non-cacheable, Outer Normal Non-cacheable outer shared (iNC-oNC-osh).
- Thus the upstream devices must present the cacheability in the *outer* cacheability attribute on PVBUS if it is cacheable. If it is a device type then both the inner and outer attributes must be set to the same. If it is iNC-oNC-osh then it must be presented as such.
- For PCIe-mode TBUs (i.e. their index appears in `list_of_pcie_mode` or `list_of_pcie_rc`):
  - Input transactions are from PCIe and the only indication of the memory type is in the NoSnoop bit of the transaction. No shareability is transported.
    - NoSnoop interpreted as iNC-oNC-osh
    - ! NoSnoop interpreted as iWB-oWB-ish (note inner shareable)
  - If a NoSnoop transaction has an attribute transform applied to it and the result of the transform is weaker than iNC-oNC-osh then it is forced to iNC-oNC-osh. For example, if a NoSnoop transaction uses a page table and is transformed to iWB-oWB-nsh then it is forced to iNC-oNC-osh. However, if the page tables transformed it to a device type then, as all device types are stronger than iNC-oNC-osh then it exits the SMMU as the device type.
  - In the model, transactions are classified by their incoming memory attributes as to whether they are NoSnoop or not and then normalized appropriately:
    - iWB-oWB-any-shareability transactions are interpreted as ! NoSnoop and therefore are normalized to iWB-oWB-ish.
    - Anything else is considered NoSnoop and therefore is normalized to iNC-oNC-osh.
  - Translated accesses also suffer the same interpretation to determine NoSnoop and how they are normalized. Thus they could enter the system with attributes different to if they were Untranslated Accesses translated by normal means.
  - It is expected that there is a component downstream of the SMMU that is aware of the system address map and will override the memory type to 'device' for any transaction that accesses a peripheral.
- The hardware has a single cacheability on input and, for transactions that are neither cache-maintenance operations nor PCIe transactions, normalizes the input to an architectural form before performing the SMMUV3 architectural transform:
  - Any device type is left untouched (the input can only represent Device-nGnRE and Device-nGnRnE).



- If the input is Write-back (WB) then it is normalized to iWB-oWB with the incoming shareability.
- If the input is anything else it is normalized to iNC-oNC-osh.
- The model accepts full architectural attributes of two levels of cacheability and so has to decide how to interpret this in terms of the hardware. For transactions that are not cache maintenance operations, the model replicates the outer attribute into the inner attribute and then performs the normalization that the hardware does.
- The hardware normalizes the architectural output attributes and outputs a single level of cacheability and a user flag (OC) specifying if the architectural attributes were cacheable in the outer cacheable domain. If the transaction is classified as a PCIe one then the NoSnoop transform described above is applied. If the original transaction was NoSnoop, then any weaker memory type is strengthened to iNC-oNC-osh so apply the following transform:

```

if      iWB-oWB-nsh/ish/osh      then output WB-nsh/ish/osh, OC = 1
else if i (NC/WT/WB) -o (WB/WT) then output NC-Sys,      OC = 1
else if i (NC/WT/WB) -oNC        then output NC-Sys,      OC = 0
else if Device- (GRE/nGRE/nGnRE) then output DV-Sys,      OC = 0
else                                output SO-Sys,      OC = 0

```

- The model only normalizes according to PCIe but otherwise leaves the architectural attributes intact on the output bus.
- MSIs are issued on the `qtw_pvbus_m` port using attributes determined by the parameter `msi_attribute_transform`, whilst Event Queue writes are always issued with `ExtendedID=0`, `UserFlags=0`, `ManagerID64=0xffffffff`.

In the hardware, there is no way of distinguishing Event Queue writes from MSI writes, however, this provides a mechanism that if the model system needs to distinguish then it can.

## Parameters for parsing transaction attributes

Some of the parameters are strings that share a common parsing format for extracting from and placing information into the transaction's attributes.

They can extract and place information into the following fields of a transaction's attributes:

- `ManagerID64` (64 bits)
- `ExtendedID` (64 bits)
- `UserFlags` (32 bits)

The string parameter is parsed as a comma-separated list of:

```
lhs_expression=rhs_expression
```

The `lhs_expression` and `rhs_expression` can be an entire symbol, for example:

```
StreamID
```

or a bit, or bit slice:

```
StreamID[16]
StreamID[31:20]
```

In `rhs_expression`, numeric constants (or slices of numeric constants) are also allowed:

```
StreamID[16]=1
StreamID[16]=10[2]
```

A single symbol might be assigned from multiple non-contiguous arrays of bits from a mix of different RHS symbols:

```
StreamID[16]=1, StreamID[31:30]=ExtendedID[1:0],
StreamID[15:0]=UserFlags[31:16], ...
```

In those cases where a left hand symbol can also appear on the right hand side, it is possible to swap bits and transform the symbol. For example, the following expression swaps bits 0 and 1 of the `ExtendedID`:

```
ExtendedID[0]=ExtendedID[1], ExtendedID[1]=ExtendedID[0]
```

Any bits of the attributes that have no transform specified are retained from the input.

The `lhs_expression` and `rhs_expression` must have the same bit width.

## Iris and MTI instances for MMU\_700

This model has the following Iris instances:

Name	Instance type
MMU_700	MMU_700
MMU_700.axi_stream_msi_manager[4294967295].pvbusmaster	PVBusMaster
MMU_700.pvbus_id_routed_m[Y].pvbusmaster (where Y = 0,4294967295)	PVBusMaster
MMU_700.register_file[0]	PVBusSlave
MMU_700.service_request_tbu[Y] (where Y = 0-63)	PVBusSlave
MMU_700.tbu[Y] (where Y = 0-63)	PVBusMapper
MMU_700.tw_msi_qs_manager[4294967295].pvbusmaster	PVBusMaster

This model has the following MTI trace components:

Name	Component type
MMU_700	MMU_700
MMU_700.axi_stream_msi_manager[4294967295].pvbusmaster	PVBusMaster
MMU_700.pvbus_id_routed_m[Y].pvbusmaster (where Y = 0,4294967295)	PVBusMaster
MMU_700.register_file[0]	PVBusSlave

Name	Component type
MMU_700.service_request_tbu[Y] (where Y = 0-63)	PVBusSlave
MMU_700.tbu[Y] (where Y = 0-63)	PVBusMapper
MMU_700.tw_msi_qs_manager[4294967295].pvbusmaster	PVBusMaster

## Ports for MMU\_700

Port	Direction	Protocol	Description
axi_stream_msi_addr_to_match_s	slave	Value_64	Any SMMU-originated MSI which exactly matches the address in this port will be sent through the AXI stream port axi_stream_msi_m which is usually connected to the GIC through axi_stream_msi_s. As MSIs are 32 bit aligned then if bits[1:0] != 0 or the address is above OAS then this is effectively disabled. NOTE that the model does not support tcu_sid[31:0] which is the MSI DeviceID to send on axi_stream_msi_m. Instead the parameter smmu_msi_device_id is used. See also the parameters: axi_stream_msi_TID and axi_stream_msi_TDEST. The default value of this port is set by the parameter axi_stream_msi_addr_to_match. This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.
axi_stream_msi_m	master	PVBus	Manager port used for sending SMMU originated MSIs directly to the GIC
clk_in	slave	ClockSignal	Clock signal (in RTL aclk) This is a clock time-base used by the TCU to spread some of its processing over time, if enabled by the wait_* parameters. The clock must always be connected.
cmd_sync_irpt_ns	master	Signal	Pulsed interrupt output signal for non-secure CMD_SYNC having a completion signal of SIG_IRQ.
cmd_sync_irpt_s	master	Signal	Pulsed interrupt output signal for secure CMD_SYNC having a completion signal of SIG_IRQ.
event_q_irpt_ns	master	Signal	Pulsed interrupt output signal for the non-secure event queue becoming non-empty.
event_q_irpt_s	master	Signal	Pulsed interrupt output signal for the secure event queue becoming non-empty.
evento	master	Signal	Event signal
global_irpt_ns	master	Signal	Pulsed interrupt output signal for non-secure SMMU_GERROR(N) signalling an error.
global_irpt_s	master	Signal	Pulsed interrupt output signal for secure SMMU_S_GERROR(N) signalling an error.
identify	master	SMMUv3AEMIdentifyProtocol	Map the transaction to the tuple (StreamID, SubStreamID, SubStreamIDValid, SSD) The StreamID that is produced by the implementation of this protocol is not the final StreamID. The final StreamID is produced by using the list_of_ns_sid_high_at_bitpos0/ list_of_s_sid_high_at_bitpos0 parameter to map the StreamID based on the upstream port index. Also see the parameter howto_identify which can replace the functionality of this port under certain circumstances.
pri_q_irpt_ns	master	Signal	Pulsed interrupt output signal for the PRI queue becoming non-empty.
prog_pvbus_s	slave	PVBus	Register subordinate port (in RTL PROG)

Port	Direction	Protocol	Description
pvbus_id_routed_m	master	PVBus	This is a special “id-routed” port for transmitting ATC invalidates upstream into the PCIe EndPoints, it is not a normal bus. See the parameter output_id_routed_transform. The FastModels ATC Invalidate and PRI Response protocol specifies how to route and deal with this port.
qtw_pvbus_m	master	PVBus	Manager port used for Table Walks, MSIs and Queue access. Note that although this is a manager port, it can still receive DVM messages.
sec_override	slave	Signal	Allow certain registers to be accessible to non-secure accesses from reset, as described in the TCU_SCR register. This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.
sup_btm	slave	Signal	System supports BTM and will be reflected in the IDR registers. This signal can override the value set by the parameters configuring the IDR registers. If BTM (Broadcast Table Maintenance) is not supported then DVM messages will be ignored. This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.
sup_cohacc	slave	Signal	System supports COHACC and will be reflected in the IDR registers. This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.
sup_httu	slave	Signal	System supports HTTU and will be reflected in the value of SMMU_IDR0.HTTU: - 0b00 (HTTU not supported) if sup_httu is 0 - 0b10 (update of AF and Dirty flags supported) if sup_httu is 1. The default value for this pin is 1. This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.
sup_sev	slave	Signal	System supports SEV and will be reflected in the IDR registers. This signal can override the value set by the parameters configuring the IDR registers. This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.
tbm_pvbus_m	master	PVBus	The TBU manager ports that carry transactions that have been translated from the correspondingly numbered tbs_pvbus_s[] port.
tbs_pvbus_s	slave	PVBus	The TBU subordinate ports that receive transactions to be translated. They will exit the SMMU through the same numbered pvbus_m[] port.
tbu_cri_irpt	master	Signal	Critical error interrupt for RAS events from the TBU. This is used for specific uncorrected errors where a System Coprocessor (SCP) might have to reset the system because the stability of the system can no longer be guaranteed. NOTE in the hardware then this is called ras_cri.
tbu_eri_irpt	master	Signal	Error Recovery Interrupt for RAS events from the TBU. This is used for uncorrected errors. NOTE in the hardware then this is called ras_eri.
tbu_fhi_irpt	master	Signal	Fault Handling Interrupt for RAS events from the TBU. Usually this is for corrected errors. NOTE in the hardware then this is called ras_fhi.
tbu_pmu_irpt	master	Signal	TBU Performance Monitoring Unit interrupt, one per TBU.
tbu_pmusnapshot_ack	master	Signal	PMU snapshot interface for the TBU, ack a snapshot.
tbu_pmusnapshot_req	slave	Signal	PMU snapshot interface for the TBU, request a snapshot.

Port	Direction	Protocol	Description
tbu_reset_in	slave	Signal	Reset signals The TBUs can have independent reset signals. However, TCU reset will result in the reset of all TBUs. This behavior is unlike SMMU RTL implementations which do allow for independent reset of the TCU separate from TBUs. Each signal tbu_reset_in[n] corresponds to the TBU using tbs_pvbus_s[n]/tbs_pvbus_s[n] pair. If the SMMU receives a transaction whilst the TBU is expected to be in reset then it will complain using the ArchMsg.Warning.warning trace source. Those tbu_reset_in that correspond to a PCIe-RC connection can be connected to monitor the PCIe-RC's reset signal. If it receives an ATS request when in reset then it will complain in a similar way. You must connect these pins if you wish the TCU_NODE_STATUS for the nodes to be accurate (including any connected to the PCIe-RC).
tcu_cri_irpt	master	Signal	Critical error interrupt for RAS events from the TCU. This is used for specific uncorrected errors where a System Coprocessor (SCP) might have to reset the system because the stability of the system can no longer be guaranteed. NOTE in the hardware then this is called ras_cri.
tcu_eri_irpt	master	Signal	Error Recovery Interrupt for RAS events from the TCU. This is used for uncorrected errors. NOTE in the hardware then this is called ras_eri.
tcu_fhi_irpt	master	Signal	Fault Handling Interrupt for RAS events from the TCU. Usually this is for corrected errors. NOTE in the hardware then this is called ras_fhi.
tcu_pmu_irpt	master	Signal	TCU Performance Monitoring Unit interrupt
tcu_pmusnapshot_ack	master	Signal	PMU snapshot interface for the TCU, ack a snapshot.
tcu_pmusnapshot_req	slave	Signal	PMU snapshot interface This is a four-phase handshake to have the corresponding PMCG perform a capture of the current counter values. PMU snapshot interface for the TCU, request a snapshot.
tcu_reset_in	slave	Signal	The reset signal to the TCU interface.

## Parameters for MMU\_700

### TCUCFG\_PARTID\_WIDTH

The width of the MPAM PARTID on the bus.

See also parameter mpam\_attribute\_transform. Accepted Values: 1 6 9.

Type: unsigned

Default value: 9

### TCUCFG\_XLATE\_SLOTS

Maximum number of outstanding stalled transactions that the SMMU supports.



TCUCFG\_XLATE\_SLOTS must be  $\geq$  TCUCFG\_PTW\_SLOTS which is currently fixed to 512.

Note

Accepted Values: 512 1024 2048 4096.

Type: `uint32_t`

Default value: 512

### **`all_error_messages_through_trace`**

Some conditions in the SMMU are so strange that the software programming the SMMU has done something wrong. At this point messages are output to either `ArchMsg.Error.*` or `ArchMsg.Warning.*` or to the error stream of the simulator. Outputting to the error stream of the simulator may cause it to return with a non-zero exit status.

If you set this option to true then instead of using the error stream of the simulator it will always use a trace stream allowing the simulation to exit with a zero exit status.

Type: `bool`

Default value: `false`

### **`axi_stream_msi_TDEST`**

ID of the AXI stream subordinate port on the GIC that receives SMMU originated MSIs sent directly. The name of the GIC port is `axi_stream_msi_s`.



If `axi_stream_msi_addr_to_match[1:0] != 0`, or the address is above OAS then this is effectively disabled.

---

See also:

- `axi_stream_msi_addr_to_match`
- `axi_stream_msi_TID`.

Type: `uint32_t`

Default value: 0

### **`axi_stream_msi_TID`**

ID of the AXI stream manager port that sends SMMU TCU originated MSIs directly to the GIC. The name of the SMMU port is `axi_stream_msi_m`.



If `axi_stream_msi_addr_to_match[1:0] != 0`, or the address is above OAS then this is effectively disabled.

---

See also:

- `axi_stream_msi_addr_to_match`
- `axi_stream_msi_TDEST`.

Type: `uint32_t`

Default value: 0

### **`axi_stream_msi_addr_to_match`**

If the last two bits are 0, any SMMU-originated MSI which exactly matches the address will be sent through the AXI stream port `axi_stream_msi_m` which is usually connected to the GIC.

This parameter drives the value of the `axi_stream_msi_addr_to_match_s` port at simulation reset. For every reset after that, the value of the port will be sampled and used if changed.



The entire address must match, including bits [1:0]. As MSIs are 32 bit aligned then if `axi_stream_msi_addr_to_match[1:0] != 0`, or the address is above OAS then this is effectively disabled.

See also:

- `axi_stream_msi_TID`
- `axi_stream_msi_TDEST`.

Type: `uint64_t`

Default value: `0xFFFFFFFFFFFFFFFF`

### **`behaviour_of_sampled_at_reset_signals`**

Some configuration signals into the SMMU are sampled on negedge of reset.

However, it can sometimes be hard to arrange to drive a configuration pin before the negedge of reset.

The configuration pins are sampled:

**0**

at negedge reset.

**1**

at negedge reset, but if a later change occurs at the same simulated time, and no transactions have occurred, then they will be resampled and the SMMU reset again.

Type: `unsigned`

Default value: 0

**cmdq\_max\_number\_of\_commands\_to\_buffer**

The command queues can buffer fetched commands before issuing them. This parameter is roughly the maximum number of commands to do this for. The programmer visible effects are that just because the CONS pointer shows a command has been consumed does not necessarily mean that it has been issued (and completed). Higher values accentuate this effect.

Type: `uint32_t`

Default value: 10

**enable\_device\_id\_checks**

If this parameter is true then the DeviceIDs seen by the GIC are:

- **for client devices**

`DeviceID = StreamID + translated_device_id_base`

- **for SMMU-generated MSIs**

`smmu_msi_device_id`

This parameter enables two checks:

- If the DeviceID is used in the `output_attribute_transform` parameter, if it overflows 32 bits then the model warns. If the DeviceID is not used, it is assumed that the external agent that forms the DeviceID warns if it overflows.
- If the SMMU supports MSIs, the model checks that the GIC is able to distinguish an MSI generated by the SMMU from one generated by a client device.

As the exact mechanism to determine the DeviceID is in the system and not necessarily under control of the SMMU then you can disable these warnings using this parameter.

See also the parameters: `output_attribute_transform` and `msi_attribute_transform`.

Type: `bool`

Default value: `true`

**howto\_identify**

If `use-identify` then the SMMU uses the `identify` port to determine the `ssd`, `StreamID`, `SubStreamID`. Otherwise, this string extracts them from the transaction's attributes.

Examples:

```
SEC_SID=ExtendedID[63], SSV=ExtendedID[62], SubstreamID=ExtendedID[51:32],
StreamID=ExtendedID[31:0]
```

or

```
nSEC_SID=ExtendedID[63], StreamID=ExtendedID[55:24], nSSV=ExtendedID[20],
SubstreamID=ExtendedID[19:0]
```



```
StreamID[31:24]=0, StreamID[23:0]=ExtendedID[23:0], SSV=1[0], ...
```

**Note**

If you are not using the `use-identify` option then the configuration string is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

LHS Symbols:

**SIDV**

Indicates that the StreamID is valid.

**SSV**

Indicates that the SubstreamID is valid.

**SEC\_SID / SEC\_SID\_bit\_1**

Bits 0 and 1 respectively of the StreamID Security State

**StreamID**

(32 b) valid if `SIDV` is 1 or both `SIDV` and `nSIDV` are unused.

**SubstreamID**

(20 b) is valid if `ssv` is true.

**nSEC\_SID / nSEC\_SID\_bit\_1 / nSSV / nSIDV**

Negative logic of above parameters. Different attributes are independent and can use negative or positive logic.

RHS Symbols:

**ExtendedID / ManagerID64 / UserFlags**

Incoming PVBUS transaction attributes

`SEC_SID` is 1b in the model. For RME systems, in hardware, then `SEC_SID` is 2b, in the model `SEC_SID_bit_1` represents bit[1].

`SIDV == 0` indicates a NoStreamID transaction and the SSD is the PAS of the transaction, `SEC_SID` is not used.

Negative and positive logic symbols for the same attribute is an error.

The model uses the term SSD (Security State Determination) to mean which security state the transaction, register, structure, event, etc. belongs to.

In the SMMUV3 Architecture, the side-band signal `SEC_SID` holds the information for the transactions, but uses a different encoding.

Before RME, `SEC_SID` was a one bit signal. With RME, in the hardware, it was expanded to 2b. To retain backwards compatibility in the model, `SEC_SID` remains as 1b in this parameter, but the second bit can be expressed with `SEC_SID_bit_1` (and its negative logic version `nSEC_SID_bit_1`)

Security state	SSD	SEC_SID_bit_1	SEC_SID
secure	0	0	1
non-secure	1	0	0
root (reserved)	2	1	1
realm	3	1	0

For those systems that support NoStreamID transactions, and `howto_identify` is not using the port, first the SMMU determines the value of `sidv` or `nsidv` to see if the transaction is a NoStreamID transaction (`sidv == 0` or `nsidv == 1`). If so then the rest of the `howto_identify` string is ignored as the information extracted relates only to StreamID transactions. Instead more information is extracted using the parameter `howto_identify_NoStreamID_extra_info`.

In AMBA systems, the equivalent signal to `sidv` is called either `ARMMUVALID` or `AWMMUVALID`. Collectively they are known as `AxMMUVALID`.

Type: `string`

Default value: `"use-identify"`

### **`ish_is_osh_DANGER`**

When set, any transaction that would use the architectural inner shareable domain is converted to use the outer shareable domain.



This parameter should match the equivalent `ish_is_osh` from the PE. If an incompatible value of the `ish_is_osh` parameter is configured for the PE and the SMMU, data coherency may be compromised.



All implementations should have this parameter as true but it is allowed in the model to give it a false value for modelling and debug purposes only.

Type: `bool`

Default value: `true`

### **`list_of_ns_sid_high_at_bitpos0`**

A comma-separated list of values to bitwise OR into each Non-secure StreamID for each TBU/Node. Bit 0 of the value corresponds to bit 0 of the StreamID.

Each TBU that is connected to a PCIe-RC (see `list_of_pcie_rc`) must serve a unique contiguous subset of StreamIDs as determined by their top bits. This is used in order to know which port to route ATC Invalidates and PRI Responses to the PCIe subsystems.

The empty string corresponds to all 0s.

Type: `string`

Default value: `""`

### **`list_of_pcie_mode`**

A comma-separated list of ranges of ports that represent TBUs that are attached to PCIe Root Complexes (PCIe-RC). A single PCIe-RC might use several TBUs and stripe accesses across them. The attribute handling for these TBUs is slightly different in that if the PCIe transaction is NoSnoop and the output attributes of the translation would be weaker than `iNC-oNC-osh` then the output is forced to `iNC-oNC-osh`.

`iNC-oNC-osh` == "inner normal non-cacheable, out normal non-cacheable, outer shared".

Type: `string`

Default value: `""`

### **`list_of_pcie_rc`**

This is a list of ports that are connected to PCIe Root Complex (PCIe-RC) by a protocol called DTI-ATS. This port is used to transport ATS and PRI Requests to the SMMU from the PCIe-RC.

In the real hardware, the PCIe-RC uses this port for ATS/PRI, and the actual transactions go through separate TBUs. In the model, this port can accept actual transactions as well. However, in the model, then the ATC Invalidates and the PRI Responses need to be transferred over the corresponding `pvbus_id_routed_m` port as DTI-ATS is bidirectional, but PVBUS is not.

Type: `string`

Default value: `""`

### **`list_of_s_sid_high_at_bitpos0`**

A comma-separated list of values to bitwise OR into each Secure StreamID for each TBU/Node. Bit 0 of the value corresponds to bit 0 of the StreamID.

The empty string corresponds to all 0s.

`use-ns` can be used to apply the `list_of_ns_sid_high_at_bitpos0` values instead.

Type: `string`

Default value: `""`

### **`mpam_attribute_transform`**



The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

If MPAM is supported, this is applied to *all* downstream transactions to transport the MPAM information.

- "" or "none" – no transform
- How to alter the output attributes. Example:

```
"ExtendedID[62:55]=MPAM_PMG, ExtendedID[54:39]=MPAM_PARTID,
ExtendedID[38]=MPAM_NS"
```

RHS/LHS Symbols:

- ExtendedID/ManagerID64/UserFlags.

RHS Symbols:

- MPAM\_PARTID
- MPAM\_PMG
- MPAM\_NS

Any bits with no transform are unchanged.



Note

- attribute transforms applied before this:
  - for client transactions 'output\_attribute\_transform'.
  - for table walks tw\_qs\_attribute\_transform.
  - for MSIs msi\_attribute\_transform.
- for translated transactions from client devices then MPAM\_NS = ! SEC\_SID.

Type: string

Default value: "ExtendedID[62:55]=MPAM\_PMG, ExtendedID[54:39]=MPAM\_PARTID, ExtendedID[38]=MPAM\_NS"

### msi\_attribute\_transform

Transform downstream attributes of MSI transactions.



Note

The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

- "" or "none" – no transform
- How to alter output attributes of SMMU-generated MSIs. Example:

```
"UserFlags[15:0]=smmu_msi_device_id[31:16],
ManagerID64[15:0]=smmu_msi_device_id[15:0],
```

```
ExtendedID=0"
```

LHS Symbols:

**ExtendedID / ManagerID64 / UserFlags**

Outgoing PVBUS transaction attributes

RHS Symbols:

**smmu\_msi\_device\_id**

The value stored in the parameter with the same name

**interrupt\_kind**

The selected bit corresponds to the interrupts listed below

**0/1**

EVENTQ s/ns

**2**

PRIQ

**3/4**

CMD\_SYNC s/ns

**5/6**

GERROR s/ns

**7/8**

PMCG s/ns

**9/10/11**

RAS FHI/ERI/CRI

**12/13**

gpf\_far/gpt\_cfg\_far

**14/15/16/17**

Realm EVENTQ/PRIQ/CMDQ/GERROR

**HWATTR\_KIND\_0**

PBHA information

**Numeric Literals**

Any number. Ex: 0x1234

ExtendedID/ManagerID64/UserFlags start with values {0, 0xFFFFffff, 0} respectively.

Any bits with no transform are unchanged.

This transform can be used to determine the DeviceID passed to the GIC to distinguish MSIs generated by the SMMU from those generated by client devices.

**Note**

See also `output_attribute_transform` and `enable_device_id_checks`.

---

**Note**

After 11.25 the `interrupt_kind` field was extended to 5 bits. This is strictly a non-backwards compatible change. However, the original 3 bits were insufficient to express all the interrupt kinds that exist.

---

Type: `string`

Default value: `"ExtendedID[31:0]=smmu_msi_device_id, ManagerID64[31:0]=0xFFFFffff"`

### **`normalize_input_normal_non_iWB_oWB_to_iNC_oNC_osh_DANGER`**

When set, use Inner Non Cacheable, Outer Non Cacheable, Outer Shareable for any upstream transaction that would use any of the following attributes:

- Normal Non-cacheable Bufferable
  - Normal Non-cacheable Non-bufferable
  - Write-through
- 

**Note**

This parameter should match the equivalent configuration from the PE. If an incompatible value of this parameter is configured for the PE and the SMMU, data coherency may be compromised.

All implementations should have this parameter as `true` but it is allowed in the model to give it a false value for modelling and debug purposes only.

---

Type: `bool`

Default value: `true`

### **`number_of_ports`**

The number of port pairs that the SMMU has.

Type: `unsigned`

Default value: `1`

### **`output_attribute_transform`**

Transform the downstream attributes of a translated transaction.



The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

- "" or "none": no transform
- How to alter output attributes. Example:

```
"ExtendedID[15:0]=DeviceID[15:0], UserFlags[31]=nSSV,
UserFlags[19:0]=SubstreamID,
ManagerID64[10]=ManagerID64[11], ManagerID64[11]=ManagerID64[10]"
```

RHS/LHS Symbols:

**ExtendedID / ManagerID64 / UserFlags**

Incoming/Outgoing PVBUS transaction attributes

RHS Symbols:

**DeviceID**

StreamID + translated\_device\_id\_base

**StreamID / SubstreamID / SEC\_SID / SSV**

Architectural information. See parameter `howto_identify` for more information.

**nSEC\_SID / nSSV**

Negative logic of above parameters. Different attributes are independent and can use negative or positive logic.

**St1PBHA / St2PBHA**

Page Based Hardware Attributes from leaf descriptors (zero if unused).

**STE\_IMPDEF1**

STE[127:116]

**HWATTR\_KIND\_0**

PBHA information

**Numeric Literals**

Any number. Ex: 0x1234

The `streamID` has had `ns_sid_high/s_sid_high` ORred into it for the appropriate TBU.



'mpam\_attribute\_transform' is applied after this.

Type: string

Default value: "ExtendedID[31:0]=DeviceID"

## output\_id\_routed\_transform



The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

The SMMU generates the following ID-routed transaction on the pvb<sub>us</sub>\_id\_routed\_m bus:

- ATC Invalidate
- PRI Response

This parameter controls how the SMMU should express:

- the StreamID
- the Trusted (T) bit

The value is a comma-separated list of assignments:

```
Address[27:12]=StreamID[15:0], ExtendedID[60]=T, ExtendedID[15:0]=StreamID[31:16]
```

Address bits[11:0] cannot be used.

The LHS can be one of:

- PAS
- ManagerID64 / ExtendedID / UserFlags
- Address

The RHS can be one of:

- a numeric constant
- SSD
- T or negative version nT
- StreamID

For realm (or 'Trusted') transactions, then `ssd=0b11`, `T=1`, `nT=0`. For non-secure (or 'Non-Trusted') transactions, then `ssd=0b01`, `T=0`, `nT=1`.

Type: string

Default value: "Address[27:12]=StreamID[15:0], PAS=SSD"



**prefetch\_only\_requests**

The simulator supports 'prefetch-only' DMI requests, which can occur at any time and for any reason and are intended to be invisible to the end execution of the model and to the user.

**0**

deny all prefetch-only requests

**1**

- use debug requests for any page table walks
  - form and use debug TLB/cache entries
  - any faults will not record, but deny the prefetch request

**2**

- treat prefetch-only requests like normal transactions
  - use normal page table walk transactions
  - use and form normal TLB/cache entries
  - faults will alter the programmer-visible state of the SMMU

0 is the safest.

1 treats the access like a debug request and requires that debug page table walks are treated correctly downstream. Any descriptors that need HTTU to allow the transaction to proceed will fail the request.

2 is dangerous, it uses real transactions and reports faults that are unphysical. Real transactions can be `wait()`ed and this disobeys the SystemC spec for `get_direct_mem_ptr()`.

Type: `unsigned`

Default value: `0`

**sec\_override**

The IMP DEF port `sec_override` controls whether some of the registers are accessible to secure or non-secure transactions. This parameter is the default value assumed for that port if the port is not driven by a signal.

Type: `bool`

Default value: `false`

**seed**

Used to seed the pseudo-random number generator that the SMMU model uses.

Type: `uint32_t`

Default value: `0x12345678`

**size\_of\_cd\_cache**

The number of entries in the cache holding CD structures. If this is zero it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

Type: `uint32_t`

Default value: 0

**size\_of\_l1cd\_cache**

The number of entries in the cache holding L1CD descriptors. If this is zero it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

Type: `uint32_t`

Default value: 0

**size\_of\_l1ste\_cache**

The number of entries in the cache holding L1STE descriptors. If this is zero it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

Type: `uint32_t`

Default value: 0

**size\_of\_ste\_cache**

The number of entries in the cache holding STE structures. If this is zero it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

Type: `uint32_t`

Default value: 0

**size\_of\_tlb**

The number of entries in the TLB. If this is zero it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

Type: `uint32_t`

Default value: 0

**smmu\_msi\_device\_id**

When appropriately enabled, assume that MSIs that are generated by the SMMU are presented to the GIC with this DeviceID.

See parameters `msi_attribute_transform` and `enable_device_id_checks`.

Type: `uint32_t`

Default value: 0

### **sup\_btm**

The default value of the register field `SMMU_IDR0.BTM` and `SMMU_ROOT_IDR0.BGPTM` (when supported). This enables Broadcast TLB maintenance.

Type: `bool`

Default value: `true`

### **sup\_cohacc**

The default value of the register `SMMU_IDR0.COHAACC`.

Type: `bool`

Default value: `true`

### **sup\_httu**

The initial value of the `sup_httu` port. See the port description for `sup_httu`.

Type: `bool`

Default value: `true`

### **sup\_oas**

The hardware has an input port `sup_oas[2:0]` that indicates what output address size (OAS) the system has. This is sampled at reset.

The model does not have this port as it is expected to be a constant for the system and not to change. Instead it is just a parameter.

The allowed values are:

- 0**  
32 bits
- 1**  
36 bits
- 2**  
40 bits
- 3**  
42 bits
- 4**  
44 bits
- 5**  
48 bits

**6**

52 bits.

Type: unsigned

Default value: 6

**sup\_sev**The default value of the register `SMMU_IDR0.SEV`.

Type: bool

Default value: true

**tlb\_when\_do\_f\_tlb\_conflict\_on\_overlap**

If a TLB entry is created by a walk and it overlaps an existing entry, there are some architectural situations where the result is known. For all others, then an implementation is allowed to use an **UNPREDICTABLE** combination of the two entries, or it can generate `F_TLB_CONFLICT`:

**0**

never generate

**1**

sometimes generate

**2**

always generate

Conflicts between global and non-global entries are not detected by the model.

Type: unsigned

Default value: 0

**translated\_device\_id\_base**

When appropriately enabled, assume that client device accesses are translated to a DeviceID as seen by the GIC of:

```
StreamID + translated_device_id_base
```

See parameter `output_attribute_transform` and `enable_device_id_checks`.

Type: uint32\_t

Default value: 0

**tw\_qs\_attribute\_transform**

Transform downstream attributes of table walk and queue transactions.



The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

- "" or "none" – no transform
- How to alter the output attributes. Example:

```
"ExtendedID[35:32]=HWATTR_KIND_0"
```

RHS/LHS Symbols:

**ExtendedID / ManagerID64 / UserFlags**

Incoming/Outgoing PVBUS transaction attributes

RHS Symbols:

**HWATTR\_KIND\_0**

PBHA information

**kind**

Transaction kind

- For a read:

**0/1**

L1STE/STE

**2/3**

L1CD/CD

**4/5**

S1/S2 TTD (including CAS)

**6**

CMDQ

**7**

VMS

**11/12**

LOGPT/L1GPT

**13/14**

LODPT/L1DPT

- For a write:

**0**

EVENTQ

**1**

PRIQ

ExtendedID/ManagerID64/UserFlags start with values {0, 0xFFFFFFFF, 0} respectively.

Any bits with no transform are unchanged.



See also `output_attribute_transform` and `msi_attribute_transform`.

---

Type: string

Default value: ""

### **version**

The version of this product.

Valid values are:

- r0p0
- r1p0.

Type: string

Default value: "r0p0"

### **wait\_cmdq\_ticks**

This is the time to wait before doing something on the command queue. If bit [32] is set 0x1\_0000\_0000 then the time waited for is a uniform randomly-distributed time  $[0, \max(2, (t \& 0xFFFFFFFF)) - 1]$ .

Type: uint64\_t

Default value: 0

### **wait\_eventq\_ticks**

This is the time to wait before doing something on the event queue. If bit [32] is set 0x1\_0000\_0000 then the time waited for is a uniform randomly-distributed time  $[0, \max(2, (t \& 0xFFFFFFFF)) - 1]$ .

Type: uint64\_t

Default value: 0

### **wait\_misc\_async\_actions\_ticks**

This is the time to wait before doing an async action that could be delayed is performed. If bit [32] is set 0x1\_0000\_0000 then the time waited for is a uniform randomly-distributed time  $[0, \max(2, (t \& 0xFFFFFFFF)) - 1]$ .

Type: uint64\_t

Default value: 0

### **wait\_msi\_ticks**

This is the time to wait before sending an MSI. If bit [32] is set 0x1\_0000\_0000 then the time waited for is a uniform randomly-distributed time  $[0, \max(2, (t \ \& \ 0xFFFFFFFF)) - 1]$ .

Type: uint64\_t

Default value: 0

### **wait\_pri\_req\_ticks**

This is the time to wait before processing a PRI Request. If bit [32] is set 0x1\_0000\_0000 then the time waited for is a uniform randomly-distributed time  $[0, \max(2, (t \ \& \ 0xFFFFFFFF)) - 1]$ .

Type: uint64\_t

Default value: 0

### **wait\_pri\_resp\_ticks**

This is the time to wait before sending a PRI Response back to the PCIe subsystem. When a PRI Response is an auto-response then the ATC might immediately make a new ATS request, that immediately fails, immediately makes a PRI Request, or auto-responds, etc. To break this loop, then we introduce a minimum time on all PRI Responses to give other components in the system a chance to run. If bit [32] is set 0x1\_0000\_0000 then the time waited for is a uniform randomly-distributed time  $[0, \max(2, (t \ \& \ 0xFFFFFFFF)) - 1]$ .

Type: uint64\_t

Default value: 1

## 3.245 MMU\_720AE

Defined in LISA/SMMU\_720AE.lisa.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

Model quality level changes:

From	To
r0p0=Preliminary support	Full support

## Limitations

- No power control
- AMBA stash operations, destructive read and destructive hint operations are not supported on PVBUS and also are not supported by the device.
- The PMU has limited functionality. Only a subset of the architecturally mandatory events are supported, as indicated by the `SMMU_PMC0_CEID0` fields. The PMU is intended for demonstration purposes only and for driver development.
- Limited RAS support
- The SYSCO interface is not implemented
- The low power interface is not implemented
- The IMP DEF MPAM register file is not implemented. This controls how the internal resources of the MMU-720AE are partitioned.
- `tcu_sid[31:0]` is not modelled, instead the parameter `smmu_msi_device_id` is used.
- The HWATTR side-band signal is available by configuring `output_attribute_transform`, `msi_attribute_transform` and `tw_qs_attribute_transform` with `HWATTR_KIND_0`. For some transactions HWATTR comes from the `SMMU_S_AGBPA[3:0]`. In the hardware, this register has an 'Update' bit[31] that should be written as 1 and will be turned to zero when all transactions using the old value have completed. The model does not implement this behavior and the Update bit is **RAZ/WI**.
- Any configuration parameter listed in the TRM but not shown in this file is not supported.
- `TCU_CTRL_AUX0-55` registers are modeled but unlike the TRM, reset values for these registers are 0. There is no functionality associated with these registers.
- There is no functionality associated with `TCU_ROOT_CTRL` register field `DIS_DVM`. Note: If `DIS_DVM` is set to 1, an error message is thrown which can be disabled by setting `all_error_messages_through_trace` to true.
- The model does not support the `eventoack` signal. In integration mode, the value of the field `eventoack` (bit 0) in `ITIN_PIU` will match the value of the `evento` signal.
- There is no functionality associated with `TCU_NODE_STATUS` register field `ATSv`, which indicates whether the node implements DTI-ATSv4.

## Notes

- A bad configuration renders the model inactive.
- Some configurations can be adjusted by configuration pins. These are only sampled at the negative edge of the reset pin. If you want to use these pins then you must drive them before sending a negative edge on the reset pin. During simulation reset the component driving them must also drive this transition again.
- Debug reads to the registers do not disturb the state.
- Writes to registers with Update flags, including debug writes, are ignored if the Update flag is already set to one.
- Debug and real accesses to the registers must be 32 bits or 64 bits.



- The model deals with groups of transactions with the same attributes and a similar range of addresses. The mapping used is remembered by the bus infrastructure and is used for subsequent sufficiently similar transactions without requiring intervention from the SMMU model, and is not traced, for instance.

The range of addresses that the mapping is valid for is determined by the SMMU model, depending on architectural and model-implementation details. However, as it is unaware of any sign-extension or the mapping that the SoC does, the SoC is responsible for subsequently narrowing the range of addresses for which this mapping is valid. Typically, this is done automatically when using PVBUSMapper.

- The hardware is a distributed SMMU and is divided into:
  - A single Translation Control Unit (TCU)
    - Has a port for the programming interface of the SMMU
    - Receives DVM messages
    - Does all the page walking, queue manipulation, etc.
  - One or more Translation Bus Units (TBUs)
    - Translate transactions from upstream (client) device into downstream transactions.
  - Zero or more connections to PCIe Root Complexes (PCIe-RCs)
  - There can be a total of 62 TBUs and PCIe-RCs attached.
  - An interconnect connecting the TBUs, PCIe-RCs to the TCU.
- A PCIe-RC has one connection to the TCU to make ATS requests but the PCIe-RC uses one or more TBUs to transform the transactions and pass them to the memory system. In the model, those TBUs are configured by the parameter `list_of_pcie_mode`. The SMMU does not know which TBUs a particular PCIe-RC is attached to.
- The TBU ORs a value into each StreamID that it receives. In the model, this is configured by the parameters:
  - `list_of_s_sid_high_at_bitpos0`
  - `list_of_ns_sid_high_at_bitpos0`
- The TCU, TBU, and the interconnect are all represented by this single model component.
- In the model, a pair of ports `tbs_pvbuss[i]/tbm_pvbuss_m[i]` represent a TBU 'i' or the `tbs_pvbuss[i]` represents an incoming connection for a PCIe-RC. The corresponding reverse connection from the TCU to the PCIe-RC is by a special bus called `pvbus_id_routed_m` that is used to transport ATC Invalidates to the PCIe-RC.
  - In order to reduce system construction complexity the `tbs_pvbuss[i]/tbm_pvbuss_m[i]` also acts as a TBU so that the PCIe-RC need not separate its normal transactions and its ATS requests.
  - However, ATC Invalidates are only sent to a port which appears in `list_of_pcie_rc`. It should be uniquely decoded to a single port based on `list_of_ns_sid_high_at_bitpos0` and those ATC Invalidates must be routed to the correct PCIe subsystem in order to invalidate the cache of ATS Response in the subsystem. Thus all TBUs that a PCIe-RC uses must have a unique reverse mapping from stream id to port.

- The pin `sup_oas` is not supported, instead it is a parameter as it is assumed that it would be tied to a fixed value in any specific platform.
- The hardware only has a single cacheability attribute for input transactions, but PVBUS transports both inner and outer cacheability.
- For non-PCIe-mode TBUs (i.e. their index does not appear in `list_of_pcie_mode` or `list_of_pcie_rc`):
  - For non-cache maintenance operations:
    - If the input attribute is any type of device then it is well defined as being outer-shared and Device-nGnRnE or Device-nGnRE. There is no support for Gathering or Reordering.
    - If the outer cacheable input attribute is normal then if it is Write-back then this is converted to Inner Write-back Outer Write-back (iWB-oWB) with the desired shareability. No Transient hint is supported and is always treated as non-transient.
    - This leaves all other normal memory types that are mapped to Inner Normal Non-cacheable, Outer Normal Non-cacheable outer shared (iNC-oNC-osh).
- Thus the upstream devices must present the cacheability in the *outer* cacheability attribute on PVBUS if it is cacheable. If it is a device type then both the inner and outer attributes must be set to the same. If it is iNC-oNC-osh then it must be presented as such.
- For PCIe-mode TBUs (i.e. their index appears in `list_of_pcie_mode` or `list_of_pcie_rc`):
  - Input transactions are from PCIe and the only indication of the memory type is in the NoSnoop bit of the transaction. No shareability is transported.
    - NoSnoop interpreted as iNC-oNC-osh
    - ! NoSnoop interpreted as iWB-oWB-ish (note inner shareable)
  - If a NoSnoop transaction has an attribute transform applied to it and the result of the transform is weaker than iNC-oNC-osh then it is forced to iNC-oNC-osh. For example, if a NoSnoop transaction uses a page table and is transformed to iWB-oWB-nsh then it is forced to iNC-oNC-osh. However, if the page tables transformed it to a device type then, as all device types are stronger than iNC-oNC-osh then it exits the SMMU as the device type.
  - In the model, transactions are classified by their incoming memory attributes as to whether they are NoSnoop or not and then normalized appropriately:
    - iWB-oWB-any-shareability transactions are interpreted as ! NoSnoop and therefore are normalized to iWB-oWB-ish.
    - Anything else is considered NoSnoop and therefore is normalized to iNC-oNC-osh.
  - Translated accesses also suffer the same interpretation to determine NoSnoop and how they are normalized. Thus they could enter the system with attributes different to if they were Untranslated Accesses translated by normal means.
  - It is expected that there is a component downstream of the SMMU that is aware of the system address map and will override the memory type to 'device' for any transaction that accesses a peripheral.
- The hardware has a single cacheability on input and, for transactions that are neither cache-maintenance operations nor PCIe transactions, normalizes the input to an architectural form before performing the SMMUV3 architectural transform:

- Any device type is left untouched (the input can only represent Device-nGnRE and Device-nGnRE).
- If the input is Write-back (WB) then it is normalized to iWB-oWB with the incoming shareability.
- If the input is anything else it is normalized to iNC-oNC-osh.
- The model accepts full architectural attributes of two levels of cacheability and so has to decide how to interpret this in terms of the hardware. For transactions that are not cache maintenance operations, the model replicates the outer attribute into the inner attribute and then performs the normalization that the hardware does.
- The hardware normalizes the architectural output attributes and outputs a single level of cacheability and a user flag (OC) specifying if the architectural attributes were cacheable in the outer cacheable domain. If the transaction is classified as a PCIe one then the NoSnoop transform described above is applied. If the original transaction was NoSnoop, then any weaker memory type is strengthened to iNC-oNC-osh so apply the following transform:

```

if      iWB-oWB-nsh/ish/osh      then output WB-nsh/ish/osh, OC = 1
else if i (NC/WT/WB) -o (WB/WT) then output NC-Sys,      OC = 1
else if i (NC/WT/WB) -oNC        then output NC-Sys,      OC = 0
else if Device-(GRE/nGRE/nGnRE) then output DV-Sys,      OC = 0
else                               output SO-Sys,      OC = 0

```

- The model only normalizes according to PCIe but otherwise leaves the architectural attributes intact on the output bus.
- MSIs are issued on the `qtw_pvbus_m` port using attributes determined by the parameter `msi_attribute_transform`, whilst Event Queue writes are always issued with `ExtendedID=0`, `UserFlags=0`, `ManagerID64=0xffffffff`.

In the hardware, there is no way of distinguishing Event Queue writes from MSI writes, however, this provides a mechanism that if the model system needs to distinguish then it can. MPAM and MEC attributes are provided by the parameters:

- `mpam_attribute_transform`
- `mec_attribute_transform` (not all versions support MEC)
- The model supports architectural features and registers matching `rOp0-00eac0`.

## Security State Determination (SSD)

The model uses the term SSD to mean the security state that the transaction, register, structure, or event belongs to. In the SMMUv3 architecture, the sideband signal `SEC_SID` holds this information for the transactions, but uses a different encoding.

Before RME, `SEC_SID` was a one-bit signal. With RME, in the hardware, it was expanded to two bits. To retain backwards compatibility in the model, `SEC_SID` remains as one-bit in the parameter `howto_identify`, but the second bit can be expressed with `SEC_SID_bit_1`, and its negative logic version `nSEC_SID_bit_1`.

Security state	SSD	SEC_SID_bit_1	SEC_SID
secure	0	0	1

Security state	SSD	SEC_SID_bit_1	SEC_SID
non-secure	1	0	0
root (reserved)	2	1	1
realm	3	1	0

Parameters for parsing transaction attributes

Some of the parameters are strings that share a common parsing format for extracting from and placing information into the transaction's attributes.

They can extract and place information into the following fields of a transaction's attributes:

- ManagerID64 (64 bits)
- ExtendedID (64 bits)
- UserFlags (32 bits)

The string parameter is parsed as a comma-separated list of:

```
lhs_expression=rhs_expression
```

The lhs\_expression and rhs\_expression can be an entire symbol, for example:

```
StreamID
```

or a bit, or bit slice:

```
StreamID[16]  
StreamID[31:20]
```

In rhs\_expression, numeric constants (or slices of numeric constants) are also allowed:

```
StreamID[16]=1  
StreamID[16]=10[2]
```

A single symbol might be assigned from multiple non-contiguous arrays of bits from a mix of different RHS symbols:

```
StreamID[16]=1, StreamID[31:30]=ExtendedID[1:0],  
StreamID[15:0]=UserFlags[31:16], ...
```

In those cases where a left hand symbol can also appear on the right hand side, it is possible to swap bits and transform the symbol. For example, the following expression swaps bits 0 and 1 of the ExtendedID:

```
ExtendedID[0]=ExtendedID[1], ExtendedID[1]=ExtendedID[0]
```

Any bits of the attributes that have no transform specified are retained from the input.

The `lhs_expression` and `rhs_expression` must have the same bit width.

## Iris and MTI instances for MMU\_720AE

This model has the following Iris instances:

Name	Instance type
MMU_720AE	MMU_720AE
MMU_720AE.axi_stream_msi_manager[4294967295].pvbusmaster	PVBusMaster
MMU_720AE.pvbus_id_routed_m[Y].pvbusmaster (where Y = 0,4294967295)	PVBusMaster
MMU_720AE.register_file[0]	PVBusSlave
MMU_720AE.service_request_tbu[Y] (where Y = 0-63)	PVBusSlave
MMU_720AE.tbu[Y] (where Y = 0-63)	PVBusMapper
MMU_720AE.tw_msi_qs_manager[4294967295].pvbusmaster	PVBusMaster

This model has the following MTI trace components:

Name	Component type
MMU_720AE	MMU_720AE
MMU_720AE.axi_stream_msi_manager[4294967295].pvbusmaster	PVBusMaster
MMU_720AE.pvbus_id_routed_m[Y].pvbusmaster (where Y = 0,4294967295)	PVBusMaster
MMU_720AE.register_file[0]	PVBusSlave
MMU_720AE.service_request_tbu[Y] (where Y = 0-63)	PVBusSlave
MMU_720AE.tbu[Y] (where Y = 0-63)	PVBusMapper
MMU_720AE.tw_msi_qs_manager[4294967295].pvbusmaster	PVBusMaster

## Ports for MMU\_720AE

Port	Direction	Protocol	Description
axi_stream_msi_addr_to_match_s	slave	Value_64	Any SMMU-originated MSI which exactly matches the address in this port will be sent through the AXI stream port <code>axi_stream_msi_m</code> which is usually connected to the GIC through <code>axi_stream_msi_s</code> . As MSIs are 32 bit aligned then if bits[1:0] != 0 or the address is above OAS then this is effectively disabled. NOTE that the model does not support <code>tcu_sid[31:0]</code> which is the MSI DeviceID to send on <code>axi_stream_msi_m</code> . Instead the parameter <code>smmu_msi_device_id</code> is used. See also the parameters: <code>axi_stream_msi_TID</code> and <code>axi_stream_msi_TDEST</code> . The default value of this port is set by the parameter <code>axi_stream_msi_addr_to_match</code> . This port is sampled at negedge of <code>tcu_reset_in</code> and must be set before the negedge of the reset signal.
axi_stream_msi_m	master	PVBus	Manager port used for sending SMMU originated MSIs directly to the GIC
clk_in	slave	ClockSignal	Clock signal (in RTL <code>adck</code> ) This is a clock time-base used by the TCU to spread some of its processing over time, if enabled by the <code>wait_*</code> parameters. The clock must always be connected.
cmd_sync_irpt_ns	master	Signal	Pulsed interrupt output signal for non-secure CMD_SYNC having a completion signal of SIG_IRQ.

Port	Direction	Protocol	Description
cmd_sync_irpt_r	master	Signal	Pulsed interrupt output signal for realm CMD_SYNC having a completion signal of SIG_IRQ. This pin exists in r0 but is tied off to 0. It is implemented in r1.
cmd_sync_irpt_s	master	Signal	Pulsed interrupt output signal for secure CMD_SYNC having a completion signal of SIG_IRQ.
event_q_irpt_ns	master	Signal	Pulsed interrupt output signal for the non-secure event queue becoming non-empty.
event_q_irpt_r	master	Signal	Pulsed interrupt output signal for the realm event queue becoming non-empty. This pin exists in r0 but is tied off to 0. It is implemented in r1.
event_q_irpt_s	master	Signal	Pulsed interrupt output signal for the secure event queue becoming non-empty.
evento	master	Signal	Event signal This aligns with the eventoreq signal on the RTL. The eventoack signal is not supported.
global_irpt_ns	master	Signal	Pulsed interrupt output signal for non-secure SMMU_GERROR(N) signalling an error.
global_irpt_r	master	Signal	Pulsed interrupt output signal for realm SMMU_R_GERROR(N) signalling an error. This pin exists in r0 but is tied off to 0. It is implemented in r1.
global_irpt_s	master	Signal	Pulsed interrupt output signal for secure SMMU_S_GERROR(N) signalling an error.
gpf_far	master	Signal	An error becomes active in SMMU_ROOT_GPF_FAR.
gpt_cfg_far	master	Signal	An error becomes active in SMMU_ROOT_GPT_CFG_FAR.
identify	master	SMMUv3AEMIdentifyProtocol	Map the transaction to the tuple (StreamID, SubStreamID, SubStreamIDValid, SSD) The StreamID that is produced by the implementation of this protocol is not the final StreamID. The final StreamID is produced by using the list_of_ns_sid_high_at_bitpos0/ list_of_s_sid_high_at_bitpos0 parameter to map the StreamID based on the upstream port index. Also see the parameter howto_identify which can replace the functionality of this port under certain circumstances.
logptsz_s	slave	Value	RME: This is a four bit signal that encodes the region size that a single LOGPT entry covers. The default value of this port is derived from the parameter rme_logpt_entry_covers_log2size_in_bytes which is in a different format to the port. If a valid value is driven then it will be put in the field SMMU_ROOT_GPT_BASE_CFG.LOGPTSZ. The port uses the same encoding as the field. If an invalid value is driven to this port and legacy_tz_en is low then the value is reported in the SMMU_ROOT_GPT_BASE_CFG.LOGPTSZ and then all transactions will fault with a GPT Configuration fault (gpt_cfg_far). This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.
legacy_tz_en	slave	Signal	Tie this high to get non-RME behaviour. On the real hardware, then each of the TCUs and the TBUs have a legacy_tz_en and they must all be driven to the same value. In the model, we only have a single version of this pin. This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.

Port	Direction	Protocol	Description
ns_gbpa_abort_init	slave	Signal	This port is an Non-secure global bypass. The ns_gbpa_abort_init signal sets the reset value of SMMU_GBPA.ABORT: 0 - On reset, do not abort all incoming transactions. 1 - On reset, abort all incoming transactions. This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.
pri_q_irpt_ns	master	Signal	Pulsed interrupt output signal for the non-secure PRI queue becoming non-empty.
pri_q_irpt_r	master	Signal	Pulsed interrupt output signal for the realm PRI queue becoming non-empty. This pin exists in r0 but is tied off to 0. It is implemented in r1.
prog_pvbus_s	slave	PVBus	Register subordinate port (in RTL PROG)
pvbus_id_routed_m	master	PVBus	This is a special "id-routed" port for transmitting ATC invalidates upstream into the PCIe EndPoints, it is not a normal bus. See parameter output_id_routed_transform. The FastModels ATC Invalidate and PRI Response protocol specifies how to route and deal with this port.
qtw_pvbus_m	master	PVBus	Manager port used for Table Walks, MSIs and Queue access. Note that although this is a manager port, it can still receive DVM messages.
s_gbpa_abort_init	slave	Signal	This port is an secure global bypass. The s_gbpa_abort_init signal sets the reset value of SMMU_S_GBPA.ABORT: 0 - On reset, do not abort all incoming transactions. 1 - On reset, abort all incoming transactions. This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.
sec_override	slave	Signal	Allow certain registers to be accessible to non-secure accesses from reset, as described in the TCU_SCR register. This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.
sup_btm	slave	Signal	System supports BTM and will be reflected in the IDR registers. This signal can override the value set by the parameters configuring the IDR registers. If BTM (Broadcast Table Maintenance) is not supported then DVM messages will be ignored. This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.
sup_cohacc	slave	Signal	System supports COHACC and will be reflected in the IDR registers. This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.
sup_httu	slave	Signal	System supports HTTU and will be reflected in the value of SMMU_IDR0.HTTU: - 0b00 (HTTU not supported) if sup_httu is 0 - 0b10 (update of AF and Dirty flags supported) if sup_httu is 1. The default value for this pin is 1. This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.
sup_sev	slave	Signal	System supports SEV and will be reflected in the IDR registers. This signal can override the value set by the parameters configuring the IDR registers. This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.
tbm_pvbus_m	master	PVBus	The TBU manager ports that carry transactions that have been translated from the correspondingly numbered tbs_pvbus_s[] port.

Port	Direction	Protocol	Description
tbs_pvbus_s	slave	PVBus	The TBU subordinate ports that receive transactions to be translated. They will exit the SMMU through the same numbered pvbus_m[] port.
tbu_crit_err	master	Signal	Critical error. This cannot occur in the model except by using an Integration Register to generate it.
tbu_fmurstdisable	slave	Signal	TBU Preserve FMU error record The TBUs have independent signals to enable a Warm Reset which preserves the state of the FMU node registers. Warm reset is enabled via the logical AND of the tbu_fmurstdisable signal and the rising edge of the tbu_reset_in signal. This signal must be asserted and stable prior to reset to enable a warm reset.
tbu_pmu_irpt	master	Signal	TBU Performance Monitoring Unit interrupt, one per TBU.
tbu_pmusnapshot_ack	master	Signal	PMU snapshot interface for the TBU, ack a snapshot.
tbu_pmusnapshot_req	slave	Signal	PMU snapshot interface for the TBU, request a snapshot.
tbu_ras_cri	master	Signal	Critical error interrupt for RAS events from the TBU. This is used for specific uncorrected errors where a System Coprocessor (SCP) might have to reset the system because the stability of the system can no longer be guaranteed. NOTE that in the MMU-720AE model this is called tbu_ras_et_cri.
tbu_ras_eri	master	Signal	Error Recovery Interrupt for RAS events from the TBU. This is used for uncorrected errors. NOTE that in the TRM of MMU-720AE model this is called tbu_ras_et_eri.
tbu_ras_fhi	master	Signal	Fault Handling Interrupt for RAS events from the TBU. Usually this is for corrected errors. NOTE that in the TRM of MMU-720AE model this is called tbu_ras_et_fhi.
tbu_ras_lt_cri	master	Signal	Level triggered critical error interrupt for RAS events from the TBU.
tbu_ras_lt_eri	master	Signal	Level triggered error recovery interrupt for RAS events from the TBU.
tbu_ras_lt_fhi	master	Signal	Level triggered fault handling interrupt for RAS events from the TBU.
tbu_ras_lt_irpt_v	master	Signal	Level triggered valid output for connection to System RAS agents. Asserted when the RAS record contains at least one valid error.
tbu_reset_in	slave	Signal	Reset signals The TBUs can have independent reset signals. However, TCU reset will result in the reset of all TBUs. This behavior is unlike SMMU RTL implementations which do allow for independent reset of the TCU separate from TBUs. Each signal tbu_reset_in[n] corresponds to the TBU using tbs_pvbus_s[n]/tbs_pvbus_m[n] pair. If the SMMU receives a transaction whilst the TBU is expected to be in reset then it will complain using the ArchMsg.Warning.warning trace source. Those tbu_reset_in that correspond to a PCIe-RC connection can be connected to monitor the PCIe-RC's reset signal. If it receives an ATS request when in reset then it will complain in a similar way. You must connect these pins if you wish the TCU_NODE_STATUS for the nodes to be accurate (including any connected to the PCIe-RC).
tcu_fmurstdisable	slave	Signal	TCU Preserve FMU error record The TCU has an independent signal to enable a Warm Reset which preserves the state of the FMU node registers. Warm reset is enabled via the logical AND of the tcu_fmurstdisable signal and the rising edge of the tcu_reset_in signal. This signal must be asserted and stable prior to reset to enable a warm reset.



Port	Direction	Protocol	Description
tcu_pmu_irpt	master	Signal	TCU Performance Monitoring Unit interrupt
tcu_pmusnapshot_ack	master	Signal	PMU snapshot interface for the TCU, ack a snapshot.
tcu_pmusnapshot_req	slave	Signal	PMU snapshot interface This is a four-phase handshake to have the corresponding PMCG perform a capture of the current counter values. PMU snapshot interface for the TCU, request a snapshot.
tcu_ras_cri	master	Signal	Critical error interrupt for RAS events from the TCU. This is used for specific uncorrected errors where a System Coprocessor (SCP) might have to reset the system because the stability of the system can no longer be guaranteed. NOTE that in the TRM of MMU-720AE model this is called tcu_ras_et_cri.
tcu_ras_eri	master	Signal	Error Recovery Interrupt for RAS events from the TCU. This is used for uncorrected errors. NOTE that in the TRM of MMU-720AE model this is called tcu_ras_et_eri.
tcu_ras_fhi	master	Signal	Fault Handling Interrupt for RAS events from the TCU. Usually this is for corrected errors. NOTE that in the TRM of MMU-720AE model this is called tcu_ras_et_fhi.
tcu_ras_lt_cri	master	Signal	Level triggered critical error interrupt for RAS events from the TCU.
tcu_ras_lt_eri	master	Signal	Level triggered error recovery interrupt for RAS events from the TCU.
tcu_ras_lt_fhi	master	Signal	Level triggered fault handling interrupt for RAS events from the TCU.
tcu_ras_lt_irpt_v	master	Signal	Level triggered valid output for connection to System RAS agents. Asserted when the RAS record contains at least one valid error.
tcu_reset_in	slave	Signal	The reset signal to the TCU interface.

## Parameters for MMU\_720AE

### TCUCFG\_DPT\_SUPPORT

MMU-S3 r1 Parameter Only: Enable Device Permission Table (DPT), a mechanism to enforce the association between granules of physical address space and the memory footprint of virtual machines.

**0**

Realm DPT is disabled

**1**

Realm DPT is enabled (default).

Type: `bool`

Default value: `true`

### TCUCFG\_DVM\_VAS

Virtual address size used by the system. Once set, this value is discoverable using `TCU_SYSDISC35.TCUCFG_DVM_VAS`.

In hardware, it is important to get this parameter correct as it determines the DVM message format. If this doesn't match the PEs, DVM messages are misinterpreted and any TLBI operations performed are incorrectly applied.

The model uses a representation of DVM that does not depend on the VA size and so misconfiguring this has no effect other than on the system discovery register value. Accepted Values: 49 53.

Type: `uint32_t`

Default value: 53

### **TCUCFG\_MECID\_WIDTH**

Memory Encryption Context (MEC) is a feature introduced in MMU-S3. The MECID is a 1-16 bit identifier that, if implemented, supports Memory Encryption Contexts for the Realm programming interface. The given value indicates the number of bits in the MECID. A value of 0 will disable MEC. Accepted Values: 0 4 8 12 16.

Type: `uint32_t`

Default value: 16

### **TCUCFG\_PARTID\_WIDTH**

The width of the MPAM PARTID on the bus.

The value 10 is just for MMU\_S3 r1.

See also parameter `mpam_attribute_transform`. Accepted Values: 1 6 9 10.

Type: `unsigned`

Default value: 9

### **TCUCFG\_XLATE\_SLOTS**

Maximum number of outstanding stalled transactions that the SMMU supports.



`TCUCFG_XLATE_SLOTS` must be  $\geq$  `TCUCFG_PTW_SLOTS` which is currently fixed to 512.

---

Accepted Values: 512 1024 2048 4096.

Type: `uint32_t`

Default value: 512

### **all\_error\_messages\_through\_trace**

Some conditions in the SMMU are so strange that the software programming the SMMU has done something wrong. At this point messages are output to either `ArchMsg.Error.*` or `ArchMsg.Warning.*` or to the error stream of the simulator. Outputting to the error stream of the simulator may cause it to return with a non-zero exit status.

If you set this option to true then instead of using the error stream of the simulator it will always use a trace stream allowing the simulation to exit with a zero exit status.

Type: `bool`

Default value: `false`

### **axi\_stream\_msi\_TDEST**

ID of the AXI stream subordinate port on the GIC that receives SMMU originated MSIs sent directly. The name of the GIC port is `axi_stream_msi_s`.



If `axi_stream_msi_addr_to_match[1:0] != 0`, or the address is above OAS then this is effectively disabled.

---

See also:

- `axi_stream_msi_addr_to_match`
- `axi_stream_msi_TID`.

Type: `uint32_t`

Default value: `0`

### **axi\_stream\_msi\_TID**

ID of the AXI stream manager port that sends SMMU TCU originated MSIs directly to the GIC. The name of the SMMU port is `axi_stream_msi_m`.



If `axi_stream_msi_addr_to_match[1:0] != 0`, or the address is above OAS then this is effectively disabled.

---

See also:

- `axi_stream_msi_addr_to_match`
- `axi_stream_msi_TDEST`.

Type: `uint32_t`

Default value: 0

### **axi\_stream\_msi\_addr\_to\_match**

If the last two bits are 0, any SMMU-originated MSI which exactly matches the address will be sent through the AXI stream port `axi_stream_msi_m` which is usually connected to the GIC.

This parameter drives the value of the `axi_stream_msi_addr_to_match_s` port at simulation reset. For every reset after that, the value of the port will be sampled and used if changed.



The entire address must match, including bits [1:0]. As MSIs are 32 bit aligned then if `axi_stream_msi_addr_to_match[1:0] != 0`, or the address is above OAS then this is effectively disabled.

---

See also:

- `axi_stream_msi_TID`
- `axi_stream_msi_TDEST`.

Type: `uint64_t`

Default value: `0xFFFFFFFFFFFFFFFF`

### **behaviour\_of\_sampled\_at\_reset\_signals**

Some configuration signals into the SMMU are sampled on negedge of reset.

However, it can sometimes be hard to arrange to drive a configuration pin before the negedge of reset.

The configuration pins are sampled:

**0**

at negedge reset.

**1**

at negedge reset, but if a later change occurs at the same simulated time, and no transactions have occurred, then they will be resampled and the SMMU reset again.

Type: `unsigned`

Default value: 0

### **cmdq\_max\_number\_of\_commands\_to\_buffer**

The command queues can buffer fetched commands before issuing them. This parameter is roughly the maximum number of commands to do this for. The programmer visible effects are that just because the CONS pointer shows a command has been consumed does not necessarily mean that it has been issued (and completed). Higher values accentuate this effect.

Type: `uint32_t`

Default value: 10

### **`enable_device_id_checks`**

If this parameter is true then the DeviceIDs seen by the GIC are:

- **for client devices**

`DeviceID = StreamID + translated_device_id_base`

- **for SMMU-generated MSIs**

`smmu_msi_device_id`

This parameter enables two checks:

- If the DeviceID is used in the `output_attribute_transform` parameter, if it overflows 32 bits then the model warns. If the DeviceID is not used, it is assumed that the external agent that forms the DeviceID warns if it overflows.
- If the SMMU supports MSIs, the model checks that the GIC is able to distinguish an MSI generated by the SMMU from one generated by a client device.

As the exact mechanism to determine the DeviceID is in the system and not necessarily under control of the SMMU then you can disable these warnings using this parameter.

See also the parameters: `output_attribute_transform` and `msi_attribute_transform`.

Type: `bool`

Default value: `true`

### **`hide_warning_ACCESSEN_GPCEN_set_to_1_in_a_single_write`**

The architecture recommends against setting `SMMU_ROOT_CR0.ACCESSEN` and `SMMU_ROOT_CR0.GPCEN` to 1 in the same write if it is possible that a concurrent client device transaction could appear at the SMMU. This is because there could be a time window where the client device transaction sees effective values `ACCESSEN == 1` and `GPCEN == 0` and so would bypass GPC checking.

If your system cannot generate this possibility then you can set this parameter to true and turn off the warning that software has set both `ACCESSEN` and `GPCEN` to 1 at the same time.

Type: `bool`

Default value: `false`

### **`hide_warning_EOPD_differs_from_what_would_be_cached`**

When this parameter is set to true, warnings that the effective EOPD value differs from what would be cached in the TLB are disabled. False (warnings are showed) by default.

Type: `bool`

Default value: `false`

**hide\_warning\_NoStreamID\_transaction\_for\_unsupported\_PAS\_or\_MPAM\_SP**

When RME is not supported then a NoStreamID transaction with `PAS[1] == 1` or `MPAM_SP[1] == 1` is treated as though `PAS[1] == 0` and `MPAM_SP[1] == 0`. This is usually a system construction error and is not expected to occur.

The SMMU warns when this occurs, but the warning can be hidden by setting this parameter.

Type: `bool`

Default value: `false`

**howto\_identify**

If `use-identify` then the SMMU uses the `identify` port to determine the `ssd`, `streamID`, `subStreamID`. Otherwise, this string extracts them from the transaction's attributes.

Examples:

```
SEC_SID=ExtendedID[63], SSV=ExtendedID[62], SubstreamID=ExtendedID[51:32],
StreamID=ExtendedID[31:0]
```

or

```
nSEC_SID=ExtendedID[63], StreamID=ExtendedID[55:24], nSSV=ExtendedID[20],
SubstreamID=ExtendedID[19:0]
StreamID[31:24]=0, StreamID[23:0]=ExtendedID[23:0], SSV=1[0], ...
```



**Note**

If you are not using the `use-identify` option then the configuration string is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

LHS Symbols:

**SIDV**

Indicates that the StreamID is valid.

**SSV**

Indicates that the SubstreamID is valid.

**SEC\_SID / SEC\_SID\_bit\_1**

Bits 0 and 1 respectively of the StreamID Security State

**StreamID**

(32 b) valid if `sidv` is 1 or both `sidv` and `nsidv` are unused.

**SubstreamID**

(20 b) is valid if `ssv` is true.

**nSEC\_SID / nSEC\_SID\_bit\_1 / nSSV / nSIDV**

Negative logic of above parameters. Different attributes are independent and can use negative or positive logic.

RHS Symbols:

**ExtendedID / ManagerID64 / UserFlags**

Incoming PVBUS transaction attributes

SEC\_SID is 1b in the model. For RME systems, in hardware, then SEC\_SID is 2b, in the model SEC\_SID\_bit\_1 represents bit[1].

SIDV == 0 indicates a NoStreamID transaction and the SSD is the PAS of the transaction, SEC\_SID is not used.

Negative and positive logic symbols for the same attribute is an error.

The model uses the term SSD (Security State Determination) to mean which security state the transaction, register, structure, event, etc. belongs to.

In the SMMUV3 Architecture, the side-band signal SEC\_SID holds the information for the transactions, but uses a different encoding.

Before RME, SEC\_SID was a one bit signal. With RME, in the hardware, it was expanded to 2b. To retain backwards compatibility in the model, SEC\_SID remains as 1b in this parameter, but the second bit can be expressed with SEC\_SID\_bit\_1 (and its negative logic version nSEC\_SID\_bit\_1)

Security state	SSD	SEC_SID_bit_1	SEC_SID
secure	0	0	1
non-secure	1	0	0
root (reserved)	2	1	1
realm	3	1	0

For those systems that support NoStreamID transactions, and `howto_identify` is not using the port, first the SMMU determines the value of `SIDV` or `nSIDV` to see if the transaction is a NoStreamID transaction (`SIDV == 0` or `nSIDV == 1`). If so then the rest of the `howto_identify` string is ignored as the information extracted relates only to StreamID transactions. Instead more information is extracted using the parameter `howto_identify_NoStreamID_extra_info`.

In AMBA systems, the equivalent signal to `SIDV` is called either `ARMMUVALID` or `AWMMUVALID`. Collectively they are known as `AxMMUVALID`.

Type: `string`

Default value: `"use-identify"`

**howto\_identify\_NoStreamID\_extra\_info**

The behavior of this parameter depends on `howto_identify`

- if it equals 'use-identify' then this must be "", otherwise there is an error.
- if it identifies a NoStreamID transaction (SIDV=0) then this parameter includes one or more of
  - MPAM\_SP
  - MPAM\_PARTID
  - MPAM\_PMG
  - MECID
  - HWATTR\_KIND\_0
- in any other case, this parameter is ignored.

Fields set in this parameter must not overlap the `sidv/nsidv` fields in `howto_identify`

Example:

```
MPAM_PMG[7:0]=ExtendedID[62:55], MPAM_PARTID[15:0]=ExtendedID[54:39],
MPAM_SP[1:0]=ExtendedID[38:37], MECID[15:0]=UserFlags[31:16]
HWATTR_KIND_0[3:0]=ExtendedID[42:39]
```



The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

Type: `string`

Default value: ""

### **ish\_is\_osh\_DANGER**

When set, any transaction that would use the architectural inner shareable domain is converted to use the outer shareable domain.



This parameter should match the equivalent `ish_is_osh` from the PE. If an incompatible value of the `ish_is_osh` parameter is configured for the PE and the SMMU, data coherency may be compromised.



All implementations should have this parameter as true but it is allowed in the model to give it a false value for modelling and debug purposes only.

Type: `bool`

Default value: `true`



**legacy\_tz\_en**

The default value of the legacy\_tz\_en pin:

**0**

RME is enabled

**1**

RME is disabled.

Type: `bool`

Default value: `false`

**list\_of\_ns\_sid\_high\_at\_bitpos0**

A comma-separated list of values to bitwise OR into each Non-secure StreamID for each TBU/Node. Bit 0 of the value corresponds to bit 0 of the StreamID.

Each TBU that is connected to a PCIe-RC (see `list_of_pcie_rc`) must serve a unique contiguous subset of StreamIDs as determined by their top bits. This is used in order to know which port to route ATC Invalidates and PRI Responses to the PCIe subsystems. The effective value for any PCIe-RC ports must be the same for non-secure and realm. See `list_of_r_sid_high_at_bitpos0`.

The empty string corresponds to all 0s.

Type: `string`

Default value: `""`

**list\_of\_pcie\_mode**

A comma-separated list of ranges of ports that represent TBUs that are attached to PCIe Root Complexes (PCIe-RC). A single PCIe-RC might use several TBUs and stripe accesses across them. The attribute handling for these TBUs is slightly different in that if the PCIe transaction is NoSnoop and the output attributes of the translation would be weaker than `INC-ONC-OSH` then the output is forced to `INC-ONC-OSH`.

`INC-ONC-OSH` == "inner normal non-cacheable, out normal non-cacheable, outer shared".

Type: `string`

Default value: `""`

**list\_of\_pcie\_rc**

This is a list of ports that are connected to PCIe Root Complex (PCIe-RC) by a protocol called DTI-ATS. This port is used to transport ATS and PRI Requests to the SMMU from the PCIe-RC.

In the real hardware, the PCIe-RC uses this port for ATS/PRI, and the actual transactions go through separate TBUs. In the model, this port can accept actual transactions as well. However, in the model, then the ATC Invalidates and the PRI Responses need to be transferred over the corresponding `pvtbus_id_routed_m` port as DTI-ATS is bidirectional, but PVBUS is not.

Type: `string`

Default value: `""`

### **`list_of_r_sid_high_at_bitpos0`**

A comma-separated list of values to bitwise OR into each Realm StreamID for each TBU/Node. Bit 0 of the value corresponds to bit 0 of the StreamID. This only has an effect for r1 and later.

Each TBU that is connected to a PCIe-RC (see `list_of_pcie_rc`) must serve a unique contiguous subset of StreamIDs as determined by their top bits. This is used in order to know which port to route ATC Invalidates and PRI Responses to the PCIe subsystems. The effective value for any PCIe-RC ports must be the same for non-secure and realm. See `list_of_ns_sid_high_at_bitpos0`.

The empty string corresponds to all 0s.

“use-ns” can be used to apply the `list_of_ns_sid_high_at_bitpos0` values instead.

Type: `string`

Default value: `""`

### **`list_of_s_sid_high_at_bitpos0`**

A comma-separated list of values to bitwise OR into each Secure StreamID for each TBU/Node. Bit 0 of the value corresponds to bit 0 of the StreamID.

The empty string corresponds to all 0s.

use-ns can be used to apply the `list_of_ns_sid_high_at_bitpos0` values instead.

Type: `string`

Default value: `""`

### **`mec_attribute_transform`**



The parameter is parsed according to the rules in the section ‘Parameters for parsing transaction attributes’.

If MEC is supported, this is applied to *all* downstream transactions to transport the MEC information.

- `""` or `“none”` – no transform
- How to alter the output attributes. Example:

```
"UserFlags[31:16]=MECID[15:0]"
```

RHS/LHS Symbols:

- ExtendedID/ManagerID64/UserFlags.

RHS Symbols:

- MECID

Any bits with no transform are unchanged.



Attribute transforms applied before this:

- for client transactions `output_attribute_transform / output_attribute_transform_for_NoStreamID`.
- for table walks `tw_qs_attribute_transform`.
- for MSIs `msi_attribute_transform`.
- if MPAM is enabled `mpam_attribute_transform`.

Type: `string`

Default value: `""`

### **`mpam_attribute_transform`**



The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

If MPAM is supported, this is applied to *all* downstream transactions to transport the MPAM information.

- `""` or `"none"` – no transform
- How to alter the output attributes. Example:

```
"ExtendedID[62:55]=MPAM_PMG, ExtendedID[54:39]=MPAM_PARTID,
ExtendedID[38]=MPAM_SP[0]"
```

RHS/LHS Symbols:

- ExtendedID/ManagerID64/UserFlags.

RHS Symbols:

- MPAM\_PARTID
- MPAM\_PMG
- MPAM\_NS

- MPAM\_SP
- numeric literals

Any bits with no transform are unchanged.



Note

- attribute transforms applied before this:
  - for client transactions `output_attribute_transform / output_attribute_transform_for_NoStreamID`.
  - for table walks `tw_qs_attribute_transform`.
  - for MSIs `msi_attribute_transform`.
- `mec_attribute_transform` is applied after this.
- for translated transactions from client devices then `MPAM_NS = ! SEC_SID`.

Type: string

Default value: "ExtendedID[62:55]=MPAM\_PMG, ExtendedID[54:39]=MPAM\_PARTID, ExtendedID[38]=MPAM\_NS"

### **msi\_attribute\_transform**

Transform downstream attributes of MSI transactions.



Note

The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

- "" or "none" – no transform
- How to alter output attributes of SMMU-generated MSIs. Example:

```
"UserFlags[15:0]=smmu_msi_device_id[31:16],
ManagerID64[15:0]=smmu_msi_device_id[15:0],
ExtendedID=0"
```

LHS Symbols:

**ExtendedID / ManagerID64 / UserFlags**

Outgoing PVBUS transaction attributes

RHS Symbols:

**smmu\_msi\_device\_id**

The value stored in the parameter with the same name

**interrupt\_kind**

The selected bit corresponds to the interrupts listed below

- 0/1  
EVENTQ s/ns
- 2  
PRIQ
- 3/4  
CMD\_SYNC s/ns
- 5/6  
GERROR s/ns
- 7/8  
PMCG s/ns
- 9/10/11  
RAS FHI/ERI/CRI
- 12/13  
gpf\_far/gpt\_cfg\_far
- 14/15/16/17  
Realm EVENTQ/PRIQ/CMDQ/GERROR

**HWATTR\_KIND\_0**  
PBHA information

**Numeric Literals**  
Any number. Ex: 0x1234

ExtendedID/ManagerID64/UserFlags start with values {0, 0xFFFFffff, 0} respectively.

Any bits with no transform are unchanged.

This transform can be used to determine the DeviceID passed to the GIC to distinguish MSIs generated by the SMMU from those generated by client devices.



See also `output_attribute_transform` and `enable_device_id_checks`.



After 11.25 the `interrupt_kind` field was extended to 5 bits. This is strictly a non-backwards compatible change. However, the original 3 bits were insufficient to express all the interrupt kinds that exist.

---

Type: `string`

Default value: `"ExtendedID[31:0]=smmu_msi_device_id, ManagerID64[31:0]=0xFFFFffff"`

**normalize\_input\_normal\_non\_iWB\_oWB\_to\_iNC\_oNC\_osh\_DANGER**

When set, use Inner Non Cacheable, Outer Non Cacheable, Outer Shareable for any upstream transaction that would use any of the following attributes:

- Normal Non-cacheable Bufferable
- Normal Non-cacheable Non-bufferable
- Write-through

**Note**

This parameter should match the equivalent configuration from the PE. If an incompatible value of this parameter is configured for the PE and the SMMU, data coherency may be compromised.

All implementations should have this parameter as true but it is allowed in the model to give it a false value for modelling and debug purposes only.

Type: `bool`

Default value: `true`

**ns\_gbpa\_abort\_init**

The default value of the tie off signal `ns_gbpa_abort_init`.

Type: `bool`

Default value: `false`

**number\_of\_ports**

The number of port pairs that the SMMU has.

Type: `unsigned`

Default value: `1`

**output\_attribute\_transform**

Transform downstream attributes of StreamID transactions.

**Note**

The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

- "" or "none": no transform
- How to alter output attributes. Example:

```
"ExtendedID[15:0]=DeviceID[15:0], UserFlags[31]=nSSV,
UserFlags[19:0]=SubstreamID,
```

```
ManagerID64[10]=ManagerID64[11], ManagerID64[11]=ManagerID64[10]"
```

RHS/LHS Symbols:

#### **ExtendedID / ManagerID64 / UserFlags**

Incoming/Outgoing PVBUS transaction attributes

RHS Symbols:

#### **DeviceID**

StreamID + translated\_device\_id\_base

#### **StreamID / SubstreamID / SEC\_SID / SEC\_SID\_bit\_1 / SIDV / SSV**

Architectural information. See parameter `howto_identify` for more information.

#### **nSEC\_SID / nSEC\_SID\_bit\_1 / nSSV / nSIDV**

Negative logic of above parameters. Different attributes are independent and can use negative or positive logic.

#### **St1PBHA / St2PBHA**

Page Based Hardware Attributes from leaf descriptors (zero if unused).

#### **STE\_IMPDEF1**

STE[127:116]

#### **HWATTR\_KIND\_0**

PBHA information

#### **Numeric Literals**

Any number. Ex: 0x1234

The `streamID` has had `ns_sid_high/s_sid_high/r_sid_high` ORred into it for the appropriate TBU.



- `mpam_attribute_transform` and `mec_attribute_transform` are applied in order after this.
- See also `output_attribute_transform_for_NoStreamID` for NoStreamID transactions.

Type: string

Default value: "ExtendedID[31:0]=DeviceID"

#### **output\_attribute\_transform\_for\_NoStreamID**



The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

Transform downstream attributes of NoStreamID transactions.

- "" or "none": no transform
- How to alter output attributes. Example:

```
"ExtendedID[15:0]=0, UserFlags[31]=1, UserFlags[19:0]=0,
ManagerID64[10]=ManagerID64[11],
ManagerID64[11]=ManagerID64[10] ManagerID64[9:6]=HWATTR_KIND_0"
```

RHS/LHS Symbols: \* ExtendedID/ManagerID64/UserFlags: incoming/outgoing attributes.

RHS Symbols: \* SIDV = 0, nSIDV = 1 (fixed values to indicate NoStreamID) \* PAS \*  
HWATTR\_KIND\_0

Any bits with no transform are unchanged.



Note

- mpam\_attribute\_transform and mec\_attribute\_transform are applied in order after this.
- see also output\_attribute\_transform for StreamID transactions.

Type: string

Default value: "ExtendedID[31:0]=0, ExtendedID[32]=1"

### output\_id\_routed\_transform



Note

The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

The SMMU generates the following ID-routed transaction on the pvbus\_id\_routed\_m bus:

- ATC Invalidate
- PRI Response

This parameter controls how the SMMU should express:

- the StreamID
- the Trusted (T) bit

The value is a comma-separated list of assignments:

```
Address[27:12]=StreamID[15:0], ExtendedID[60]=T, ExtendedID[15:0]=StreamID[31:16]
```

Address bits[11:0] cannot be used.

The LHS can be one of:



- PAS
- ManagerID64 / ExtendedID / UserFlags
- Address

The RHS can be one of:

- a numeric constant
- SSD
- T or negative version nT
- StreamID

For realm (or 'Trusted') transactions, then `ssd=0b11`, `T=1`, `nT=0`. For non-secure (or 'Non-Trusted') transactions, then `ssd=0b01`, `T=0`, `nT=1`.

Type: `string`

Default value: `"Address[27:12]=StreamID[15:0], PAS=SSD"`

### **prefetch\_only\_requests**

The simulator supports 'prefetch-only' DMI requests, which can occur at any time and for any reason and are intended to be invisible to the end execution of the model and to the user.

**0**

deny all prefetch-only requests

**1**

- use debug requests for any page table walks
  - form and use debug TLB/cache entries
  - any faults will not record, but deny the prefetch request

**2**

- treat prefetch-only requests like normal transactions
  - use normal page table walk transactions
  - use and form normal TLB/cache entries
  - faults will alter the programmer-visible state of the SMMU

0 is the safest.

1 treats the access like a debug request and requires that debug page table walks are treated correctly downstream. Any descriptors that need HTTU to allow the transaction to proceed will fail the request.

2 is dangerous, it uses real transactions and reports faults that are unphysical. Real transactions can be `wait()`ed and this disobeys the SystemC spec for `get_direct_mem_ptr()`.

Type: `unsigned`

Default value: 0

### **rme\_l0gpt\_entry\_covers\_log2size\_in\_bytes**

Each LOGPT entry covers:

```
2**rme_l0gpt_entry_covers_log2size_in_bytes
```

bytes of address space.

The valid values for this parameter are: 30, 34, 36, 39

This parameter is reported in an encoded format as the read-only field:

```
SMMU_ROOT_GPT_BASE_CFG.LOGPTSZ
```

This parameter can be overridden by the port `l0gptsz_s` when sampled on negedge of reset.

Type: `uint32_t`

Default value: 30

### **s\_gbpa\_abort\_init**

The default value of the tie off signal `s_gbpa_abort_init`.

Type: `bool`

Default value: `false`

### **sec\_override**

The IMP DEF port `sec_override` controls whether some of the registers are accessible to secure or non-secure/realm transactions. This parameter is the default value assumed for that port if the port is not driven by a signal.

Type: `bool`

Default value: `false`

### **seed**

Used to seed the pseudo-random number generator that the SMMU model uses.

Type: `uint32_t`

Default value: `0x12345678`

### **size\_of\_cd\_cache**

The number of entries in the cache holding CD structures. If this is zero it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

Type: `uint32_t`

Default value: 0

### **`size_of_dpttlb`**

The number of entries in the DPT TLB. If this is zero then it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

Type: `uint32_t`

Default value: 0

### **`size_of_gpttlb`**

The number of entries in the GPT TLB. If this is zero then it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

Type: `uint32_t`

Default value: 0

### **`size_of_l1cd_cache`**

The number of entries in the cache holding L1CD descriptors. If this is zero it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

Type: `uint32_t`

Default value: 0

### **`size_of_l1ste_cache`**

The number of entries in the cache holding L1STE descriptors. If this is zero it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

Type: `uint32_t`

Default value: 0

### **`size_of_ste_cache`**

The number of entries in the cache holding STE structures. If this is zero it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

Type: `uint32_t`

Default value: 0

### **`size_of_tlb`**

The number of entries in the TLB. If this is zero it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

Type: `uint32_t`

Default value: 0

### **`smmu_msi_device_id`**

When appropriately enabled, assume that MSIs that are generated by the SMMU are presented to the GIC with this DeviceID.

See parameters `msi_attribute_transform` and `enable_device_id_checks`.

Type: `uint32_t`

Default value: 0

### **`sup_btm`**

The default value of the register field `SMMU_IDR0.BTM` and `SMMU_ROOT_IDR0.BGPTM` (when supported). This enables Broadcast TLB maintenance.

Type: `bool`

Default value: `true`

### **`sup_cohacc`**

The default value of the register `SMMU_IDR0.COHAACC`.

Type: `bool`

Default value: `true`

### **`sup_httu`**

The initial value of the `sup_httu` port. See the port description for `sup_httu`.

Type: `bool`

Default value: `true`

### **`sup_oas`**

The hardware has an input port `sup_oas[2:0]` that indicates what output address size (OAS) the system has. This is sampled at reset.

The model does not have this port as it is expected to be a constant for the system and not to change. Instead it is just a parameter.

The allowed values are:

**0**

32 bits

- 1**  
36 bits
- 2**  
40 bits
- 3**  
42 bits
- 4**  
44 bits
- 5**  
48 bits
- 6**  
52 bits.

Type: `unsigned`

Default value: 6

### **`sup_sev`**

The default value of the register `SMMU_IDR0.SEV`.

Type: `bool`

Default value: `true`

### **`tlb_when_do_f_tlb_conflict_on_overlap`**

If a TLB entry is created by a walk and it overlaps an existing entry, there are some architectural situations where the result is known. For all others, then an implementation is allowed to use an **UNPREDICTABLE** combination of the two entries, or it can generate `F_TLB_CONFLICT`:

- 0**  
never generate
- 1**  
sometimes generate
- 2**  
always generate

Conflicts between global and non-global entries are not detected by the model.

Type: `unsigned`

Default value: 0

**translated\_device\_id\_base**

When appropriately enabled, assume that client device accesses are translated to a DeviceID as seen by the GIC of:

```
StreamID + translated_device_id_base
```

See parameter `output_attribute_transform` and `enable_device_id_checks`.

Type: `uint32_t`

Default value: 0

**tw\_qs\_attribute\_transform**

Transform downstream attributes of table walk and queue transactions.



Note

The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

- "" or "none" – no transform
- How to alter the output attributes. Example:

```
"ExtendedID[35:32]=HWATTR_KIND_0"
```

RHS/LHS Symbols:

**ExtendedID / ManagerID64 / UserFlags**

Incoming/Outgoing PVBUS transaction attributes

RHS Symbols:

**HWATTR\_KIND\_0**

PBHA information

**kind**

Transaction kind

- For a read:
  - 0/1** L1STE/STE
  - 2/3** L1CD/CD
  - 4/5** S1/S2 TTD (including CAS)

**6**

CMDQ

**7**

VMS

**11/12**

LOGPT/L1GPT

**13/14**

L0DPT/L1DPT

- For a write:

**0**

EVENTQ

**1**

PRIQ

ExtendedID/ManagerID64/UserFlags start with values {0, 0xFFFFffff, 0} respectively.

Any bits with no transform are unchanged.



See also `output_attribute_transform` and `msi_attribute_transform`.

Type: `string`

Default value: `""`

### **version**

The version of this product.

Valid values are:

- `r0p0`
- `r1p0`.

Type: `string`

Default value: `"r0p0"`

### **wait\_cmdq\_ticks**

This is the time to wait before doing something on the command queue. If bit [32] is set `0x1_0000_0000` then the time waited for is a uniform randomly-distributed time `[0, max(2, (t & 0xFFFFffff)) - 1]`.

Type: `uint64_t`

Default value: 0

### **`wait_eventq_ticks`**

This is the time to wait before doing something on the event queue. If bit [32] is set `0x1_0000_0000` then the time waited for is a uniform randomly-distributed time  $[0, \max(2, (t \ \& \ 0xFFFFffff)) - 1]$ .

Type: `uint64_t`

Default value: 0

### **`wait_misc_async_actions_ticks`**

This is the time to wait before doing an async action that could be delayed is performed. If bit [32] is set `0x1_0000_0000` then the time waited for is a uniform randomly-distributed time  $[0, \max(2, (t \ \& \ 0xFFFFffff)) - 1]$ .

Type: `uint64_t`

Default value: 0

### **`wait_msi_ticks`**

This is the time to wait before sending an MSI. If bit [32] is set `0x1_0000_0000` then the time waited for is a uniform randomly-distributed time  $[0, \max(2, (t \ \& \ 0xFFFFffff)) - 1]$ .

Type: `uint64_t`

Default value: 0

### **`wait_pri_req_ticks`**

This is the time to wait before processing a PRI Request. If bit [32] is set `0x1_0000_0000` then the time waited for is a uniform randomly-distributed time  $[0, \max(2, (t \ \& \ 0xFFFFffff)) - 1]$ .

Type: `uint64_t`

Default value: 0

### **`wait_pri_resp_ticks`**

This is the time to wait before sending a PRI Response back to the PCIe subsystem. When a PRI Response is an auto-response then the ATC might immediately make a new ATS request, that immediately fails, immediately makes a PRI Request, or auto-responds, etc. To break this loop, then we introduce a minimum time on all PRI Responses to give other components in the system a chance to run. If bit [32] is set `0x1_0000_0000` then the time waited for is a uniform randomly-distributed time  $[0, \max(2, (t \ \& \ 0xFFFFffff)) - 1]$ .

Type: `uint64_t`

Default value: 1



## 3.246 MMU\_L1

Defined in `LISA/SMMU_L1.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support
r0p1	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Limitations

- No power control
- AMBA stash operations, destructive read and destructive hint operations are not supported on PVBUS and also are not supported by the device.
- The PMU has limited functionality. Only a subset of the architecturally mandatory events are supported, as indicated by the `SMMU_PMCG_CEID0` fields. The PMU is intended for demonstration purposes only and for driver development.
- Limited RAS support
- The SYSCO interface is not implemented
- The low power interface is not implemented
- The IMP DEF MPAM register file is not implemented. This controls how the internal resources of the MMU\_L1 are partitioned.
- `tcu_sid[31:0]` is not modelled, instead the parameter `smmu_msi_device_id` is used.
- The HWATTR side-band signal is available by configuring `output_attribute_transform`, `msi_attribute_transform` and `tw_qs_attribute_transform` with `HWATTR_KIND_0`. For some transactions HWATTR comes from the `SMMU_S_AGBPA[3:0]`. In the hardware, this register has an 'Update' bit[31] that should be written as 1 and will be turned to zero when all transactions using the old value have completed. The model does not implement this behavior and the Update bit is **RAZ/WI**.
- Any configuration parameter listed in the TRM but not shown in this file is not supported.
- `TCU_STATUS.GNT_XLATE_SLOTS` always reads at 512

### Notes

- A bad configuration renders the model inactive.
- Some configurations can be adjusted by configuration pins. These are only sampled at the negative edge of the reset pin. If you want to use these pins then you must drive them before sending a negative edge on the reset pin. During simulation reset the component driving them must also drive this transition again.
- Debug reads to the registers do not disturb the state.

- Writes to registers with Update flags, including debug writes, are ignored if the Update flag is already set to one.
- Debug and real accesses to the registers must be 32 bits or 64 bits.
- The model deals with groups of transactions with the same attributes and a similar range of addresses. The mapping used is remembered by the bus infrastructure and is used for subsequent sufficiently similar transactions without requiring intervention from the SMMU model, and is not traced, for instance.

The range of addresses that the mapping is valid for is determined by the SMMU model, depending on architectural and model-implementation details. However, as it is unaware of any sign-extension or the mapping that the SoC does, the SoC is responsible for subsequently narrowing the range of addresses for which this mapping is valid. Typically, this is done automatically when using PVBUSMapper.

- The hardware is a distributed SMMU and is divided into:
  - A single Translation Control Unit (TCU)
    - Has a port for the programming interface of the SMMU
    - Receives DVM messages
    - Does all the page walking, queue manipulation, etc.
  - One or more Translation Bus Units (TBUs)
    - Translate transactions from upstream (client) device into downstream transactions.
  - Zero or more connections to PCIe Root Complexes (PCIe-RCs)
  - There can be a total of 62 TBUs and PCIe-RCs attached.
  - An interconnect connecting the TBUs, PCIe-RCs to the TCU.
- A PCIe-RC has one connection to the TCU to make ATS requests but the PCIe-RC uses one or more TBUs to transform the transactions and pass them to the memory system. In the model, those TBUs are configured by the parameter `list_of_pcie_mode`. The SMMU does not know which TBUs a particular PCIe-RC is attached to.
- The TBU ORs a value into each StreamID that it receives. In the model, this is configured by the parameters:
  - `list_of_s_sid_high_at_bitpos0`
  - `list_of_ns_sid_high_at_bitpos0`
- The TCU, TBU, and the interconnect are all represented by this single model component.
- In the model, a pair of ports `tbs_pvbus_s[i]/tbm_pvbus_m[i]` represent a TBU 'i' or the `tbs_pvbus_s[i]` represents an incoming connection for a PCIe-RC. The corresponding reverse connection from the TCU to the PCIe-RC is by a special bus called `pvbus_id_routed_m` that is used to transport ATC Invalidates to the PCIe-RC.
  - In order to reduce system construction complexity the `tbs_pvbus_s[i]/tbm_pvbus_m[i]` also acts as a TBU so that the PCIe-RC need not separate its normal transactions and its ATS requests.
  - However, ATC Invalidates are only sent to a port which appears in `list_of_pcie_rc`. It should be uniquely decoded to a single port based on `list_of_ns_sid_high_at_bitpos0`

and those ATC Invalidates must be routed to the correct PCIe subsystem in order to invalidate the cache of ATS Response in the subsystem. Thus all TBUs that a PCIe-RC uses must have a unique reverse mapping from stream id to port.

- The pin `sup_oas` is not supported, instead it is a parameter as it is assumed that it would be tied to a fixed value in any specific platform.
- The hardware only has a single cacheability attribute for input transactions, but PVBUS transports both inner and outer cacheability.
- For non-PCIe-mode TBUs (i.e. their index does not appear in `list_of_pcie_mode` or `list_of_pcie_rc`):
  - For non-cache maintenance operations:
    - If the input attribute is any type of device then it is well defined as being outer-shared and Device-nGnRnE or Device-nGnRE. There is no support for Gathering or Reordering.
    - If the outer cacheable input attribute is normal then if it is Write-back then this is converted to Inner Write-back Outer Write-back (iWB-oWB) with the desired shareability. No Transient hint is supported and is always treated as non-transient.
    - This leaves all other normal memory types that are mapped to Inner Normal Non-cacheable, Outer Normal Non-cacheable outer shared (iNC-oNC-osh).
- Thus the upstream devices must present the cacheability in the *outer* cacheability attribute on PVBUS if it is cacheable. If it is a device type then both the inner and outer attributes must be set to the same. If it is iNC-oNC-osh then it must be presented as such.
- For PCIe-mode TBUs (i.e. their index appears in `list_of_pcie_mode` or `list_of_pcie_rc`):
  - Input transactions are from PCIe and the only indication of the memory type is in the NoSnoop bit of the transaction. No shareability is transported.
    - NoSnoop interpreted as iNC-oNC-osh
    - ! NoSnoop interpreted as iWB-oWB-ish (note inner shareable)
  - If a NoSnoop transaction has an attribute transform applied to it and the result of the transform is weaker than iNC-oNC-osh then it is forced to iNC-oNC-osh. For example, if a NoSnoop transaction uses a page table and is transformed to iWB-oWB-nsh then it is forced to iNC-oNC-osh. However, if the page tables transformed it to a device type then, as all device types are stronger than iNC-oNC-osh then it exits the SMMU as the device type.
  - In the model, transactions are classified by their incoming memory attributes as to whether they are NoSnoop or not and then normalized appropriately:
    - iWB-oWB-any-shareability transactions are interpreted as ! NoSnoop and therefore are normalized to iWB-oWB-ish.
    - Anything else is considered NoSnoop and therefore is normalized to iNC-oNC-osh.
  - Translated accesses also suffer the same interpretation to determine NoSnoop and how they are normalized. Thus they could enter the system with attributes different to if they were Untranslated Accesses translated by normal means.
  - It is expected that there is a component downstream of the SMMU that is aware of the system address map and will override the memory type to 'device' for any transaction that accesses a peripheral.

- The hardware has a single cacheability on input and, for transactions that are neither cache-maintenance operations nor PCIe transactions, normalizes the input to an architectural form before performing the SMMUv3 architectural transform:
  - Any device type is left untouched (the input can only represent Device-nGnRE and Device-nGnRnE).
  - If the input is Write-back (WB) then it is normalized to iWB-oWB with the incoming shareability.
  - If the input is anything else it is normalized to iNC-oNC-osh.
- The model accepts full architectural attributes of two levels of cacheability and so has to decide how to interpret this in terms of the hardware. For transactions that are not cache maintenance operations, the model replicates the outer attribute into the inner attribute and then performs the normalization that the hardware does.
- The hardware normalizes the architectural output attributes and outputs a single level of cacheability and a user flag (OC) specifying if the architectural attributes were cacheable in the outer cacheable domain. If the transaction is classified as a PCIe one then the NoSnoop transform described above is applied. If the original transaction was NoSnoop, then any weaker memory type is strengthened to iNC-oNC-osh so apply the following transform:

```

if      iWB-oWB-nsh/ish/osh      then output WB-nsh/ish/osh, OC = 1
else if i (NC/WT/WB) -o (WB/WT)  then output NC-Sys,      OC = 1
else if i (NC/WT/WB) -oNC        then output NC-Sys,      OC = 0
else if Device-(GRE/nGRE/nGnRE) then output DV-Sys,      OC = 0
else                               output SO-Sys,      OC = 0

```

- The model only normalizes according to PCIe but otherwise leaves the architectural attributes intact on the output bus.
- MSIs are issued on the `qtw_pvbus_m` port using attributes determined by the parameter `msi_attribute_transform`, whilst Event Queue writes are always issued with `ExtendedID=0`, `UserFlags=0`, `ManagerID64=0xFFFFfff`.

In the hardware, there is no way of distinguishing Event Queue writes from MSI writes, however, this provides a mechanism that if the model system needs to distinguish then it can.

- The model supports architectural features and registers matching `rOp1-00eac0`.

## Parameters for parsing transaction attributes

Some of the parameters are strings that share a common parsing format for extracting from and placing information into the transaction's attributes.

They can extract and place information into the following fields of a transaction's attributes:

- `ManagerID64` (64 bits)
- `ExtendedID` (64 bits)
- `UserFlags` (32 bits)

The string parameter is parsed as a comma-separated list of:

```
lhs_expression=rhs_expression
```

The `lhs_expression` and `rhs_expression` can be an entire symbol, for example:

```
StreamID
```

or a bit, or bit slice:

```
StreamID[16]  
StreamID[31:20]
```

In `rhs_expression`, numeric constants (or slices of numeric constants) are also allowed:

```
StreamID[16]=1  
StreamID[16]=10[2]
```

A single symbol might be assigned from multiple non-contiguous arrays of bits from a mix of different RHS symbols:

```
StreamID[16]=1, StreamID[31:30]=ExtendedID[1:0],  
StreamID[15:0]=UserFlags[31:16], ...
```

In those cases where a left hand symbol can also appear on the right hand side, it is possible to swap bits and transform the symbol. For example, the following expression swaps bits 0 and 1 of the `ExtendedID`:

```
ExtendedID[0]=ExtendedID[1], ExtendedID[1]=ExtendedID[0]
```

Any bits of the attributes that have no transform specified are retained from the input.

The `lhs_expression` and `rhs_expression` must have the same bit width.

## Iris and MTI instances for MMU\_L1

This model has the following Iris instances:

Name	Instance type
MMU_L1	MMU_L1
MMU_L1.axi_stream_msi_manager[4294967295].pvbusmaster	PVBusMaster
MMU_L1.pvbus_id_routed_m[Y].pvbusmaster (where Y = 0,4294967295)	PVBusMaster
MMU_L1.register_file[0]	PVBusSlave
MMU_L1.service_request_tbu[Y] (where Y = 0-63)	PVBusSlave
MMU_L1.tbu[Y] (where Y = 0-63)	PVBusMapper
MMU_L1.tw_msi_qs_manager[4294967295].pvbusmaster	PVBusMaster

This model has the following MTI trace components:

Name	Component type
MMU_L1	MMU_L1
MMU_L1.axi_stream_msi_manager[4294967295].pvbusmaster	PVBusMaster
MMU_L1.pvbus_id_routed_m[Y].pvbusmaster (where Y = 0,4294967295)	PVBusMaster
MMU_L1.register_file[0]	PVBusSlave
MMU_L1.service_request_tbu[Y] (where Y = 0-63)	PVBusSlave
MMU_L1.tbu[Y] (where Y = 0-63)	PVBusMapper
MMU_L1.tw_msi_qs_manager[4294967295].pvbusmaster	PVBusMaster

### Ports for MMU\_L1

Port	Direction	Protocol	Description
axi_stream_msi_addr_to_match_s	slave	Value_64	Any SMMU-originated MSI which exactly matches the address in this port will be sent through the AXI stream port axi_stream_msi_m which is usually connected to the GIC through axi_stream_msi_s. As MSIs are 32 bit aligned then if bits[1:0] != 0 or the address is above OAS then this is effectively disabled. NOTE that the model does not support tcu_sid[31:0] which is the MSI DeviceID to send on axi_stream_msi_m. Instead the parameter smmu_msi_device_id is used. See also the parameters: axi_stream_msi_TID and axi_stream_msi_TDEST. The default value of this port is set by the parameter axi_stream_msi_addr_to_match. This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.
axi_stream_msi_m	master	PVBus	Manager port used for sending SMMU originated MSIs directly to the GIC
clk_in	slave	ClockSignal	Clock signal (in RTL ack) This is a clock time-base used by the TCU to spread some of its processing over time, if enabled by the wait_* parameters. The clock must always be connected.
cmd_sync_irpt_ns	master	Signal	Pulsed interrupt output signal for non-secure CMD_SYNC having a completion signal of SIG_IRQ.
cmd_sync_irpt_s	master	Signal	Pulsed interrupt output signal for secure CMD_SYNC having a completion signal of SIG_IRQ.
event_q_irpt_ns	master	Signal	Pulsed interrupt output signal for the non-secure event queue becoming non-empty.
event_q_irpt_s	master	Signal	Pulsed interrupt output signal for the secure event queue becoming non-empty.
evento	master	Signal	Event signal
global_irpt_ns	master	Signal	Pulsed interrupt output signal for non-secure SMMU_GERROR(N) signalling an error.
global_irpt_s	master	Signal	Pulsed interrupt output signal for secure SMMU_S_GERROR(N) signalling an error.

Port	Direction	Protocol	Description
identify	master	SMMUv3AEMIdentifyProtocol	Map the transaction to the tuple (StreamID, SubStreamID, SubStreamIDValid, SSD) The StreamID that is produced by the implementation of this protocol is not the final StreamID. The final StreamID is produced by using the list_of_ns_sid_high_at_bitpos0/ list_of_s_sid_high_at_bitpos0 parameter to map the StreamID based on the upstream port index. Also see the parameter howto_identify which can replace the functionality of this port under certain circumstances.
pri_q_irpt_ns	master	Signal	Pulsed interrupt output signal for the PRI queue becoming non-empty.
prog_pvbus_s	slave	PVBus	Register subordinate port (in RTL PROG)
pvbus_id_routed_m	master	PVBus	This is a special “id-routed” port for transmitting ATC invalidates upstream into the PCIe EndPoints, it is not a normal bus. The FastModels ATC Invalidate and PRI Response protocol specifies how to route and deal with this port.
qtw_pvbus_m	master	PVBus	Manager port used for Table Walks, MSIs and Queue access. Note that although this is a manager port, it can still receive DVM messages.
sec_override	slave	Signal	Allow certain registers to be accessible to non-secure accesses from reset, as described in the TCU_SCR register. This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.
sup_btm	slave	Signal	System supports BTM and will be reflected in the IDR registers. This signal can override the value set by the parameters configuring the IDR registers. If BTM (Broadcast Table Maintenance) is not supported then DVM messages will be ignored. This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.
sup_cohacc	slave	Signal	System supports COHACC and will be reflected in the IDR registers. This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.
sup_httu	slave	Signal	System supports HTTU and will be reflected in the value of SMMU_IDR0.HTTU: - 0b00 (HTTU not supported) if sup_httu is 0 - 0b10 (update of AF and Dirty flags supported) if sup_httu is 1 The default value for this pin is 1. This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.
sup_sev	slave	Signal	System supports SEV and will be reflected in the IDR registers. This signal can override the value set by the parameters configuring the IDR registers. This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.
tbm_pvbus_m	master	PVBus	The TBU manager ports that carry transactions that have been translated from the correspondingly numbered tbs_pvbus_s[] port.
tbs_pvbus_s	slave	PVBus	The TBU subordinate ports that receive transactions to be translated. They will exit the SMMU through the same numbered pvbus_m[] port.
tbu_pmu_irpt	master	Signal	TBU Performance Monitoring Unit interrupt, one per TBU.
tbu_pmusnapshot_ack	master	Signal	PMU snapshot interface for the TBU, ack a snapshot.
tbu_pmusnapshot_req	slave	Signal	PMU snapshot interface for the TBU, request a snapshot.

Port	Direction	Protocol	Description
tbu_ras_cri	master	Signal	Critical Error Interrupt for RAS events from the TBU. This is used for specific uncorrected errors where a System Coprocessor (SCP) might have to reset the system because the stability of the system can no longer be guaranteed. NOTE in the hardware then this is called ras_cri.
tbu_ras_eri	master	Signal	Error Recovery Interrupt for RAS events from the TBU. This is used for uncorrected errors. NOTE in the hardware then this is called ras_eri.
tbu_ras_fhi	master	Signal	Fault Handling Interrupt for RAS events from the TBU. Usually this is for corrected errors. NOTE in the hardware then this is called ras_fhi.
tbu_reset_in	slave	Signal	Reset signals The TBUs can have independent reset signals. However, TCU reset will result in the reset of all TBUs. This behavior is unlike SMMU RTL implementations which do allow for independent reset of the TCU separate from TBUs. Each signal tbu_reset_in[n] corresponds to the TBU using tbs_pvbus_s[n]/tbs_pvbus_s[n] pair. If the SMMU receives a transaction whilst the TBU is expected to be in reset then it will complain using the ArchMsg.Warning.warning trace source. Those tbu_reset_in that correspond to a PCIe-RC connection can be connected to monitor the PCIe-RC's reset signal. If it receives an ATS request when in reset then it will complain in a similar way. You must connect these pins if you wish the TCU_NODE_STATUS for the nodes to be accurate (including any connected to the PCIe-RC).
tcu_pmu_irpt	master	Signal	TCU Performance Monitoring Unit interrupt
tcu_pmusnapshot_ack	master	Signal	PMU snapshot interface for the TCU, ack a snapshot.
tcu_pmusnapshot_req	slave	Signal	PMU snapshot interface This is a four-phase handshake to have the corresponding PMCG perform a capture of the current counter values. PMU snapshot interface for the TCU, request a snapshot.
tcu_ras_cri	master	Signal	Critical Error Interrupt for RAS events from the TCU. This is used for specific uncorrected errors where a System Coprocessor (SCP) might have to reset the system because the stability of the system can no longer be guaranteed. NOTE in the hardware then this is called ras_cri.
tcu_ras_eri	master	Signal	Error Recovery Interrupt for RAS events from the TCU. This is used for uncorrected errors. NOTE in the hardware then this is called ras_eri.
tcu_ras_fhi	master	Signal	Fault Handling Interrupt for RAS events from the TCU. Usually this is for corrected errors. NOTE in the hardware then this is called ras_fhi.
tcu_reset_in	slave	Signal	The reset signal to the TCU interface.

## Parameters for MMU\_L1

### TCUCFG\_PARTID\_WIDTH

The width of the MPAM PARTID on the bus.

See also parameter mpam\_attribute\_transform. Accepted Values: 1 6 9.

Type: unsigned



Default value: 9

### **TCUCFG\_XLATE\_SLOTS**

Maximum number of outstanding stalled transactions that the SMMU supports.



TCUCFG\_XLATE\_SLOTS must be  $\geq$  TCUCFG\_PTW\_SLOTS which is currently fixed to 512.

---

Accepted Values: 512 1024 2048 4096.

Type: `uint32_t`

Default value: 512

### **all\_error\_messages\_through\_trace**

Some conditions in the SMMU are so strange that the software programming the SMMU has done something wrong. At this point messages are output to either `ArchMsg.Error.*` or `ArchMsg.Warning.*` or to the error stream of the simulator. Outputting to the error stream of the simulator may cause it to return with a non-zero exit status.

If you set this option to true then instead of using the error stream of the simulator it will always use a trace stream allowing the simulation to exit with a zero exit status.

Type: `bool`

Default value: `false`

### **axi\_stream\_msi\_TDEST**

ID of the AXI stream subordinate port on the GIC that receives SMMU originated MSIs sent directly. The name of the GIC port is `axi_stream_msi_s`.



If `axi_stream_msi_addr_to_match[1:0] != 0`, or the address is above OAS then this is effectively disabled.

---

See also:

- `axi_stream_msi_addr_to_match`
- `axi_stream_msi_TID`.

Type: `uint32_t`

Default value: 0

### **axi\_stream\_msi\_TID**

ID of the AXI stream manager port that sends SMMU TCU originated MSIs directly to the GIC. The name of the SMMU port is `axi_stream_msi_m`.



Note

If `axi_stream_msi_addr_to_match[1:0] != 0`, or the address is above OAS then this is effectively disabled.

See also:

- `axi_stream_msi_addr_to_match`
- `axi_stream_msi_TDEST`.

Type: `uint32_t`

Default value: 0

### **axi\_stream\_msi\_addr\_to\_match**

If the last two bits are 0, any SMMU-originated MSI which exactly matches the address will be sent through the AXI stream port `axi_stream_msi_m` which is usually connected to the GIC.

This parameter drives the value of the `axi_stream_msi_addr_to_match_s` port at simulation reset. For every reset after that, the value of the port will be sampled and used if changed.



Note

The entire address must match, including bits [1:0]. As MSIs are 32 bit aligned then if `axi_stream_msi_addr_to_match[1:0] != 0`, or the address is above OAS then this is effectively disabled.

See also:

- `axi_stream_msi_TID`
- `axi_stream_msi_TDEST`.

Type: `uint64_t`

Default value: `0xFFFFFFFFFFFFFFFF`

### **behaviour\_of\_sampled\_at\_reset\_signals**

Some configuration signals into the SMMU are sampled on negedge of reset.

However, it can sometimes be hard to arrange to drive a configuration pin before the negedge of reset.

The configuration pins are sampled:

**0**

at negedge reset.

**1**

at negedge reset, but if a later change occurs at the same simulated time, and no transactions have occurred, then they will be resampled and the SMMU reset again.

Type: unsigned

Default value: 0

**cmdq\_max\_number\_of\_commands\_to\_buffer**

The command queues can buffer fetched commands before issuing them. This parameter is roughly the maximum number of commands to do this for. The programmer visible effects are that just because the CONS pointer shows a command has been consumed does not necessarily mean that it has been issued (and completed). Higher values accentuate this effect.

Type: uint32\_t

Default value: 10

**enable\_device\_id\_checks**

If this parameter is true then the DeviceIDs seen by the GIC are:

- **for client devices**

$$\text{DeviceID} = \text{StreamID} + \text{translated\_device\_id\_base}$$

- **for SMMU-generated MSIs**

$$\text{smmu\_msi\_device\_id}$$

This parameter enables two checks:

- If the DeviceID is used in the `output_attribute_transform` parameter, if it overflows 32 bits then the model warns. If the DeviceID is not used, it is assumed that the external agent that forms the DeviceID warns if it overflows.
- If the SMMU supports MSIs, the model checks that the GIC is able to distinguish an MSI generated by the SMMU from one generated by a client device.

As the exact mechanism to determine the DeviceID is in the system and not necessarily under control of the SMMU then you can disable these warnings using this parameter.

See also the parameters: `output_attribute_transform` and `msi_attribute_transform`.

Type: bool

Default value: true

**howto\_identify**

If `use-identify` then the SMMU uses the `identify` port to determine the `ssd`, `streamID`, `subStreamID`. Otherwise, this string extracts them from the transaction's attributes.

Examples:

```
SEC_SID=ExtendedID[63], SSV=ExtendedID[62], SubstreamID=ExtendedID[51:32],
StreamID=ExtendedID[31:0]
```

or

```
nSEC_SID=ExtendedID[63], StreamID=ExtendedID[55:24], nSSV=ExtendedID[20],
SubstreamID=ExtendedID[19:0]
StreamID[31:24]=0, StreamID[23:0]=ExtendedID[23:0], SSV=1[0], ...
```



If you are not using the `use-identify` option then the configuration string is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

LHS Symbols:

**SIDV**

Indicates that the StreamID is valid.

**SSV**

Indicates that the SubstreamID is valid.

**SEC\_SID / SEC\_SID\_bit\_1**

Bits 0 and 1 respectively of the StreamID Security State

**StreamID**

(32 b) valid if `SIDV` is 1 or both `SIDV` and `nSIDV` are unused.

**SubstreamID**

(20 b) is valid if `ssv` is true.

**nSEC\_SID / nSEC\_SID\_bit\_1 / nSSV / nSIDV**

Negative logic of above parameters. Different attributes are independent and can use negative or positive logic.

RHS Symbols:

**ExtendedID / ManagerID64 / UserFlags**

Incoming PVBUS transaction attributes

`SEC_SID` is 1b in the model. For RME systems, in hardware, then `SEC_SID` is 2b, in the model `SEC_SID_bit_1` represents bit[1].

`SIDV == 0` indicates a NoStreamID transaction and the SSD is the PAS of the transaction, `SEC_SID` is not used.

Negative and positive logic symbols for the same attribute is an error.

The model uses the term SSD (Security State Determination) to mean which security state the transaction, register, structure, event, etc. belongs to.

In the SMMUV3 Architecture, the side-band signal `SEC_SID` holds the information for the transactions, but uses a different encoding.

Before RME, `SEC_SID` was a one bit signal. With RME, in the hardware, it was expanded to 2b. To retain backwards compatibility in the model, `SEC_SID` remains as 1b in this parameter, but the second bit can be expressed with `SEC_SID_bit_1` (and its negative logic version `nSEC_SID_bit_1`)

Security state	SSD	SEC_SID_bit_1	SEC_SID
secure	0	0	1
non-secure	1	0	0
root (reserved)	2	1	1
realm	3	1	0

For those systems that support NoStreamID transactions, and `howto_identify` is not using the port, first the SMMU determines the value of `SIDV` or `nSIDV` to see if the transaction is a NoStreamID transaction (`SIDV == 0` or `nSIDV == 1`). If so then the rest of the `howto_identify` string is ignored as the information extracted relates only to StreamID transactions. Instead more information is extracted using the parameter `howto_identify_NoStreamID_extra_info`.


In AMBA systems, the equivalent signal to `SIDV` is called either `ARMMUVALID` OR `AWMMUVALID`. Collectively they are known as `AxMMUVALID`.

Type: `string`

Default value: `"use-identify"`


**ish\_is\_osh\_DANGER**

When set, any transaction that would use the architectural inner shareable domain is converted to use the outer shareable domain.



Note

This parameter should match the equivalent `ish_is_osh` from the PE. If an incompatible value of the `ish_is_osh` parameter is configured for the PE and the SMMU, data coherency may be compromised.



Note

All implementations should have this parameter as true but it is allowed in the model to give it a false value for modelling and debug purposes only.

Type: `bool`

Default value: `true`

**list\_of\_ns\_sid\_high\_at\_bitpos0**

A comma-separated list of values to bitwise OR into each Non-secure StreamID for each TBU/Node. Bit 0 of the value corresponds to bit 0 of the StreamID.

Each TBU that is connected to a PCIe-RC (see `list_of_pcie_rc`) must serve a unique contiguous subset of StreamIDs as determined by their top bits. This is used in order to know which port to route ATC Invalidates and PRI Responses to the PCIe subsystems.

The empty string corresponds to all 0s.

Type: `string`

Default value: ""

**list\_of\_pcie\_mode**

A comma-separated list of ranges of ports that represent TBUs that are attached to PCIe Root Complexes (PCIe-RC). A single PCIe-RC might use several TBUs and stripe accesses across them. The attribute handling for these TBUs is slightly different in that if the PCIe transaction is NoSnoop and the output attributes of the translation would be weaker than `inc-unc-osh` then the output is forced to `inc-unc-osh`.

`inc-unc-osh` == "inner normal non-cacheable, out normal non-cacheable, outer shared".

Type: `string`

Default value: ""

**list\_of\_pcie\_rc**

This is a list of ports that are connected to PCIe Root Complex (PCIe-RC) by a protocol called DTI-ATS. This port is used to transport ATS and PRI Requests to the SMMU from the PCIe-RC.

In the real hardware, the PCIe-RC uses this port for ATS/PRI, and the actual transactions go through separate TBUs. In the model, this port can accept actual transactions as well. However, in the model, then the ATC Invalidates and the PRI Responses need to be transferred over the corresponding `pvtbus_id_routed_m` port as DTI-ATS is bidirectional, but PVBUS is not.

Type: `string`

Default value: ""

**list\_of\_s\_sid\_high\_at\_bitpos0**

A comma-separated list of values to bitwise OR into each Secure StreamID for each TBU/Node. Bit 0 of the value corresponds to bit 0 of the StreamID.

The empty string corresponds to all 0s.

`use-ns` can be used to apply the `list_of_ns_sid_high_at_bitpos0` values instead.

Type: string

Default value: ""

## mpam\_attribute\_transform



The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

If MPAM is supported, this is applied to *all* downstream transactions to transport the MPAM information.

- "" or "none" – no transform
- How to alter the output attributes. Example:

```
"ExtendedID[62:55]=MPAM_PMG, ExtendedID[54:39]=MPAM_PARTID,
ExtendedID[38]=MPAM_NS"
```

RHS/LHS Symbols:

- ExtendedID/ManagerID64/UserFlags.

RHS Symbols:

- MPAM\_PARTID
- MPAM\_PMG
- MPAM\_NS

Any bits with no transform are unchanged.



- attribute transforms applied before this:
  - for client transactions 'output\_attribute\_transform'.
  - for table walks tw\_qs\_attribute\_transform.
  - for MSIs msi\_attribute\_transform.
- for translated transactions from client devices then MPAM\_NS = ! SEC\_SID.

Type: string

Default value: "ExtendedID[62:55]=MPAM\_PMG, ExtendedID[54:39]=MPAM\_PARTID, ExtendedID[38]=MPAM\_NS"

## msi\_attribute\_transform

Transform downstream attributes of MSI transactions.



The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

- "" or "none" – no transform
- How to alter output attributes of SMMU-generated MSIs. Example:

```
"UserFlags[15:0]=smmu_msi_device_id[31:16],
ManagerID64[15:0]=smmu_msi_device_id[15:0],
ExtendedID=0"
```

LHS Symbols:

**ExtendedID / ManagerID64 / UserFlags**

Outgoing PVBUS transaction attributes

RHS Symbols:

**smmu\_msi\_device\_id**

The value stored in the parameter with the same name

**interrupt\_kind**

The selected bit corresponds to the interrupts listed below

**0/1**

EVENTQ s/ns

**2**

PRIQ

**3/4**

CMD\_SYNC s/ns

**5/6**

GERROR s/ns

**7/8**

PMCG s/ns

**9/10/11**

RAS FHI/ERI/CRI

**12/13**

gpf\_far/gpt\_cfg\_far

**14/15/16/17**

Realm EVENTQ/PRIQ/CMDQ/GERROR

**HWATTR\_KIND\_0**

PBHA information



## Numeric Literals

Any number. Ex: 0x1234

ExtendedID/ManagerID64/UserFlags start with values {0, 0xFFFFFFFF, 0} respectively.

Any bits with no transform are unchanged.

This transform can be used to determine the DeviceID passed to the GIC to distinguish MSIs generated by the SMMU from those generated by client devices.



Note

See also `output_attribute_transform` and `enable_device_id_checks`.



Note

After 11.25 the `interrupt_kind` field was extended to 5 bits. This is strictly a non-backwards compatible change. However, the original 3 bits were insufficient to express all the interrupt kinds that exist.

Type: `string`

Default value: `"ExtendedID[31:0]=smmu_msi_device_id, ManagerID64[31:0]=0xFFFFFFFF"`

### **`normalize_input_normal_non_iWB_oWB_to_iNC_oNC_osh_DANGER`**

When set, use Inner Non Cacheable, Outer Non Cacheable, Outer Shareable for any upstream transaction that would use any of the following attributes:

- Normal Non-cacheable Bufferable
- Normal Non-cacheable Non-bufferable
- Write-through



Note

This parameter should match the equivalent configuration from the PE. If an incompatible value of this parameter is configured for the PE and the SMMU, data coherency may be compromised.

All implementations should have this parameter as true but it is allowed in the model to give it a false value for modelling and debug purposes only.

Type: `bool`

Default value: `true`

### **`number_of_ports`**

The number of port pairs that the SMMU has.

Type: unsigned

Default value: 1

### output\_attribute\_transform

Transform the downstream attributes of a translated transaction.



The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

- "" or "none": no transform
- How to alter output attributes. Example:

```
"ExtendedID[15:0]=DeviceID[15:0], UserFlags[31]=nSSV,
UserFlags[19:0]=SubstreamID,
ManagerID64[10]=ManagerID64[11], ManagerID64[11]=ManagerID64[10]"
```

RHS/LHS Symbols:

#### **ExtendedID / ManagerID64 / UserFlags**

Incoming/Outgoing PVBUS transaction attributes

RHS Symbols:

#### **DeviceID**

StreamID + translated\_device\_id\_base

#### **StreamID / SubstreamID / SEC\_SID / SSV**

Architectural information. See parameter `howto_identify` for more information.

#### **nSEC\_SID / nSSV**

Negative logic of above parameters. Different attributes are independent and can use negative or positive logic.

#### **St1PBHA / St2PBHA**

Page Based Hardware Attributes from leaf descriptors (zero if unused).

#### **STE\_IMPDEF1**

STE[127:116]

#### **HWATTR\_KIND\_0**

PBHA information

#### **Numeric Literals**

Any number. Ex: 0x1234

The `streamID` has had `ns_sid_high/s_sid_high` ORred into it for the appropriate TBU.

**Note**

'mpam\_attribute\_transform' is applied after this.

Type: `string`

Default value: `"ExtendedID[31:0]=DeviceID"`

### **output\_id\_routed\_transform**

**Note**

The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

The SMMU generates the following ID-routed transaction on the `pvbus_id_routed_m` bus:

- ATC Invalidate
- PRI Response

This parameter controls how the SMMU should express:

- the StreamID
- the Trusted (T) bit

The value is a comma-separated list of assignments:

```
Address[27:12]=StreamID[15:0], ExtendedID[60]=T, ExtendedID[15:0]=StreamID[31:16]
```

Address bits[11:0] cannot be used.

The LHS can be one of:

- PAS
- ManagerID64 / ExtendedID / UserFlags
- Address

The RHS can be one of:

- a numeric constant
- SSD
- T or negative version nT
- StreamID

For realm (or 'Trusted') transactions, then  $ssd=0b11$ ,  $T=1$ ,  $nT=0$ . For non-secure (or 'Non-Trusted') transactions, then  $ssd=0b01$ ,  $T=0$ ,  $nT=1$ .

Type: `string`

Default value: `"Address[27:12]=StreamID[15:0], PAS=SSD"`

### **prefetch\_only\_requests**

The simulator supports 'prefetch-only' DMI requests, which can occur at any time and for any reason and are intended to be invisible to the end execution of the model and to the user.

**0**

deny all prefetch-only requests

**1**

- use debug requests for any page table walks
  - form and use debug TLB/cache entries
  - any faults will not record, but deny the prefetch request

**2**

- treat prefetch-only requests like normal transactions
  - use normal page table walk transactions
  - use and form normal TLB/cache entries
  - faults will alter the programmer-visible state of the SMMU

0 is the safest.

1 treats the access like a debug request and requires that debug page table walks are treated correctly downstream. Any descriptors that need HTTU to allow the transaction to proceed will fail the request.

2 is dangerous, it uses real transactions and reports faults that are unphysical. Real transactions can be `wait()`ed and this disobeys the SystemC spec for `get_direct_mem_ptr()`.

Type: `unsigned`

Default value: 0

### **sec\_override**

The IMP DEF port `sec_override` controls whether some of the registers are accessible to secure or non-secure transactions. This parameter is the default value assumed for that port if the port is not driven by a signal.

Type: `bool`

Default value: `false`

**seed**

Used to seed the pseudo-random number generator that the SMMU model uses.

Type: `uint32_t`

Default value: `0x12345678`

**size\_of\_cd\_cache**

The number of entries in the cache holding CD structures. If this is zero it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

Type: `uint32_t`

Default value: `0`

**size\_of\_l1cd\_cache**

The number of entries in the cache holding L1CD descriptors. If this is zero it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

Type: `uint32_t`

Default value: `0`

**size\_of\_l1ste\_cache**

The number of entries in the cache holding L1STE descriptors. If this is zero it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

Type: `uint32_t`

Default value: `0`

**size\_of\_ste\_cache**

The number of entries in the cache holding STE structures. If this is zero it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

Type: `uint32_t`

Default value: `0`

**size\_of\_tlb**

The number of entries in the TLB. If this is zero it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

Type: `uint32_t`

Default value: `0`

**smmu\_msi\_device\_id**

When appropriately enabled, assume that MSIs that are generated by the SMMU are presented to the GIC with this DeviceID.

See parameters `msi_attribute_transform` and `enable_device_id_checks`.

Type: `uint32_t`

Default value: `0`

**sup\_btm**

The default value of the register field `SMMU_IDR0.BTM` and `SMMU_ROOT_IDR0.BGPTM` (when supported). This enables Broadcast TLB maintenance.

Type: `bool`

Default value: `true`

**sup\_cohacc**

The default value of the register `SMMU_IDR0.COACC`.

Type: `bool`

Default value: `true`

**sup\_httu**

The initial value of the `sup_httu` port. See the port description for `sup_httu`.

Type: `bool`

Default value: `true`

**sup\_oas**

The hardware has an input port `sup_oas[2:0]` that indicates what output address size (OAS) the system has. This is sampled at reset.

The model does not have this port as it is expected to be a constant for the system and not to change. Instead it is just a parameter.

The allowed values are:

**0**

32 bits

**1**

36 bits

**2**

40 bits

**3**

42 bits

**4**

44 bits

**5**

48 bits

**6**

52 bits.

Type: unsigned

Default value: 6

**sup\_sev**The default value of the register `SMMU_IDR0.SEV`.

Type: bool

Default value: true

**tlb\_when\_do\_f\_tlb\_conflict\_on\_overlap**

If a TLB entry is created by a walk and it overlaps an existing entry, there are some architectural situations where the result is known. For all others, then an implementation is allowed to use an **UNPREDICTABLE** combination of the two entries, or it can generate `F_TLB_CONFLICT`:

**0**

never generate

**1**

sometimes generate

**2**

always generate

Conflicts between global and non-global entries are not detected by the model.

Type: unsigned

Default value: 0

**translated\_device\_id\_base**

When appropriately enabled, assume that client device accesses are translated to a DeviceID as seen by the GIC of:

```
StreamID + translated_device_id_base
```

See parameter `output_attribute_transform` and `enable_device_id_checks`.

Type: `uint32_t`

Default value: 0

### **tw\_qs\_attribute\_transform**

Transform downstream attributes of table walk and queue transactions.



The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

- "" or "none" – no transform
- How to alter the output attributes. Example:

```
"ExtendedID[35:32]=HWATTR_KIND_0"
```

RHS/LHS Symbols:

**ExtendedID / ManagerID64 / UserFlags**

Incoming/Outgoing PVBUS transaction attributes

RHS Symbols:

**HWATTR\_KIND\_0**

PBHA information

**kind**

Transaction kind

- For a read:

**0/1**

L1STE/STE

**2/3**

L1CD/CD

**4/5**

S1/S2 TTD (including CAS)

**6**

CMDQ

**7**

VMS

**11/12**

LOGPT/L1GPT

**13/14**

LODPT/L1DPT



- For a write:
 

<b>0</b>	EVENTQ
<b>1</b>	PRIQ

ExtendedID/ManagerID64/UserFlags start with values {0, 0xFFFFFFFF, 0} respectively.

Any bits with no transform are unchanged.



See also `output_attribute_transform` and `msi_attribute_transform`.

---

Type: string

Default value: ""

### **version**

The version of this product.

Valid values are:

- r0p0.

Type: string

Default value: "r0p0"

### **wait\_cmdq\_ticks**

This is the time to wait before doing something on the command queue. If bit [32] is set 0x1\_0000\_0000 then the time waited for is a uniform randomly-distributed time  $[0, \max(2, (t \& 0xFFFFFFFF)) - 1]$ .

Type: uint64\_t

Default value: 0

### **wait\_eventq\_ticks**

This is the time to wait before doing something on the event queue. If bit [32] is set 0x1\_0000\_0000 then the time waited for is a uniform randomly-distributed time  $[0, \max(2, (t \& 0xFFFFFFFF)) - 1]$ .

Type: uint64\_t

Default value: 0

**wait\_misc\_async\_actions\_ticks**

This is the time to wait before doing an async action that could be delayed is performed. If bit [32] is set 0x1\_0000\_0000 then the time waited for is a uniform randomly-distributed time  $[0, \max(2, (t \& 0xFFFFFFFF)) - 1]$ .

Type: uint64\_t

Default value: 0

**wait\_msi\_ticks**

This is the time to wait before sending an MSI. If bit [32] is set 0x1\_0000\_0000 then the time waited for is a uniform randomly-distributed time  $[0, \max(2, (t \& 0xFFFFFFFF)) - 1]$ .

Type: uint64\_t

Default value: 0

**wait\_pri\_req\_ticks**

This is the time to wait before processing a PRI Request. If bit [32] is set 0x1\_0000\_0000 then the time waited for is a uniform randomly-distributed time  $[0, \max(2, (t \& 0xFFFFFFFF)) - 1]$ .

Type: uint64\_t

Default value: 0

**wait\_pri\_resp\_ticks**

This is the time to wait before sending a PRI Response back to the PCIe subsystem. When a PRI Response is an auto-response then the ATC might immediately make a new ATS request, that immediately fails, immediately makes a PRI Request, or auto-responds, etc. To break this loop, then we introduce a minimum time on all PRI Responses to give other components in the system a chance to run. If bit [32] is set 0x1\_0000\_0000 then the time waited for is a uniform randomly-distributed time  $[0, \max(2, (t \& 0xFFFFFFFF)) - 1]$ .

Type: uint64\_t

Default value: 1

3.247 MMU\_S3

Defined in LISA/SMMU\_S3.lisa.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support
r1p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

## Limitations

- No power control
- AMBA stash operations, destructive read and destructive hint operations are not supported on PVBUS and also are not supported by the device.
- The PMU has limited functionality. Only a subset of the architecturally mandatory events are supported, as indicated by the `SMMU_PMCG_CEID0` fields. The PMU is intended for demonstration purposes only and for driver development.
- Limited RAS support
- The SYSCO interface is not implemented
- The low power interface is not implemented
- The IMP DEF MPAM register file is not implemented. This controls how the internal resources of the MMU-S3 are partitioned.
- `tcu_sid[31:0]` is not modelled, instead the parameter `smmu_msi_device_id` is used.
- The HWATTR side-band signal is available by configuring `output_attribute_transform`, `msi_attribute_transform` and `tw_qs_attribute_transform` with `HWATTR_KIND_0`. For some transactions HWATTR comes from the `SMMU_S_AGBPA[3:0]`. In the hardware, this register has an 'Update' bit[31] that should be written as 1 and will be turned to zero when all transactions using the old value have completed. The model does not implement this behavior and the Update bit is **RAZ/WI**.
- Any configuration parameter listed in the TRM but not shown in this file is not supported.
- `TCU_CTRL_AUX0-55` registers are modeled but unlike the TRM, reset values for these registers are 0. There is no functionality associated with these registers.
- There is no functionality associated with `TCU_ROOT_CTRL` register field `DIS_DVM`. Note: If `DIS_DVM` is set to 1, an error message is thrown which can be disabled by setting `all_error_messages_through_trace` to true.
- The model does not support the `eventoack` signal. In integration mode, the value of the field `eventoack` (bit 0) in `ITIN_PIU` will match the value of the `evento` signal.
- There is no functionality associated with `TCU_NODE_STATUS` register field `ATSv`, which indicates whether the node implements DTI-ATSv4.

## Notes

- A bad configuration renders the model inactive.
- Some configurations can be adjusted by configuration pins. These are only sampled at the negative edge of the reset pin. If you want to use these pins then you must drive them before sending a negative edge on the reset pin. During simulation reset the component driving them must also drive this transition again.
- Debug reads to the registers do not disturb the state.
- Writes to registers with Update flags, including debug writes, are ignored if the Update flag is already set to one.

- Debug and real accesses to the registers must be 32 bits or 64 bits.
- The model deals with groups of transactions with the same attributes and a similar range of addresses. The mapping used is remembered by the bus infrastructure and is used for subsequent sufficiently similar transactions without requiring intervention from the SMMU model, and is not traced, for instance.

The range of addresses that the mapping is valid for is determined by the SMMU model, depending on architectural and model-implementation details. However, as it is unaware of any sign-extension or the mapping that the SoC does, the SoC is responsible for subsequently narrowing the range of addresses for which this mapping is valid. Typically, this is done automatically when using PVBUSMapper.

- The hardware is a distributed SMMU and is divided into:
  - A single Translation Control Unit (TCU)
    - Has a port for the programming interface of the SMMU
    - Receives DVM messages
    - Does all the page walking, queue manipulation, etc.
  - One or more Translation Bus Units (TBUs)
    - Translate transactions from upstream (client) device into downstream transactions.
  - Zero or more connections to PCIe Root Complexes (PCIe-RCs)
  - There can be a total of 62 TBUs and PCIe-RCs attached.
  - An interconnect connecting the TBUs, PCIe-RCs to the TCU.
- A PCIe-RC has one connection to the TCU to make ATS requests but the PCIe-RC uses one or more TBUs to transform the transactions and pass them to the memory system. In the model, those TBUs are configured by the parameter `list_of_pcie_mode`. The SMMU does not know which TBUs a particular PCIe-RC is attached to.
- The TBU ORs a value into each StreamID that it receives. In the model, this is configured by the parameters:
  - `list_of_s_sid_high_at_bitpos0`
  - `list_of_ns_sid_high_at_bitpos0`
- The TCU, TBU, and the interconnect are all represented by this single model component.
- In the model, a pair of ports `tbs_pvbuss[i]/tbm_pvbuss_m[i]` represent a TBU 'i' or the `tbs_pvbuss[i]` represents an incoming connection for a PCIe-RC. The corresponding reverse connection from the TCU to the PCIe-RC is by a special bus called `pvbus_id_routed_m` that is used to transport ATC Invalidates to the PCIe-RC.
  - In order to reduce system construction complexity the `tbs_pvbuss[i]/tbm_pvbuss_m[i]` also acts as a TBU so that the PCIe-RC need not separate its normal transactions and its ATS requests.
  - However, ATC Invalidates are only sent to a port which appears in `list_of_pcie_rc`. It should be uniquely decoded to a single port based on `list_of_ns_sid_high_at_bitpos0` and those ATC Invalidates must be routed to the correct PCIe subsystem in order to invalidate the cache of ATS Response in the subsystem. Thus all TBUs that a PCIe-RC uses must have a unique reverse mapping from stream id to port.

- The pin `sup_oas` is not supported, instead it is a parameter as it is assumed that it would be tied to a fixed value in any specific platform.
- The hardware only has a single cacheability attribute for input transactions, but PVBUS transports both inner and outer cacheability.
- For non-PCIe-mode TBUs (i.e. their index does not appear in `list_of_pcie_mode` or `list_of_pcie_rc`):
  - For non-cache maintenance operations:
    - If the input attribute is any type of device then it is well defined as being outer-shared and Device-nGnRnE or Device-nGnRE. There is no support for Gathering or Reordering.
    - If the outer cacheable input attribute is normal then if it is Write-back then this is converted to Inner Write-back Outer Write-back (iWB-oWB) with the desired shareability. No Transient hint is supported and is always treated as non-transient.
    - This leaves all other normal memory types that are mapped to Inner Normal Non-cacheable, Outer Normal Non-cacheable outer shared (iNC-oNC-osh).
- Thus the upstream devices must present the cacheability in the *outer* cacheability attribute on PVBUS if it is cacheable. If it is a device type then both the inner and outer attributes must be set to the same. If it is iNC-oNC-osh then it must be presented as such.
- For PCIe-mode TBUs (i.e. their index appears in `list_of_pcie_mode` or `list_of_pcie_rc`):
  - Input transactions are from PCIe and the only indication of the memory type is in the NoSnoop bit of the transaction. No shareability is transported.
    - NoSnoop interpreted as iNC-oNC-osh
    - ! NoSnoop interpreted as iWB-oWB-ish (note inner shareable)
  - If a NoSnoop transaction has an attribute transform applied to it and the result of the transform is weaker than iNC-oNC-osh then it is forced to iNC-oNC-osh. For example, if a NoSnoop transaction uses a page table and is transformed to iWB-oWB-nsh then it is forced to iNC-oNC-osh. However, if the page tables transformed it to a device type then, as all device types are stronger than iNC-oNC-osh then it exits the SMMU as the device type.
  - In the model, transactions are classified by their incoming memory attributes as to whether they are NoSnoop or not and then normalized appropriately:
    - iWB-oWB-any-shareability transactions are interpreted as ! NoSnoop and therefore are normalized to iWB-oWB-ish.
    - Anything else is considered NoSnoop and therefore is normalized to iNC-oNC-osh.
  - Translated accesses also suffer the same interpretation to determine NoSnoop and how they are normalized. Thus they could enter the system with attributes different to if they were Untranslated Accesses translated by normal means.
  - It is expected that there is a component downstream of the SMMU that is aware of the system address map and will override the memory type to 'device' for any transaction that accesses a peripheral.
- The hardware has a single cacheability on input and, for transactions that are neither cache-maintenance operations nor PCIe transactions, normalizes the input to an architectural form before performing the SMMUV3 architectural transform:

- Any device type is left untouched (the input can only represent Device-nGnRE and Device-nGnRnE).
- If the input is Write-back (WB) then it is normalized to iWB-oWB with the incoming shareability.
- If the input is anything else it is normalized to iNC-oNC-osh.
- The model accepts full architectural attributes of two levels of cacheability and so has to decide how to interpret this in terms of the hardware. For transactions that are not cache maintenance operations, the model replicates the outer attribute into the inner attribute and then performs the normalization that the hardware does.
- The hardware normalizes the architectural output attributes and outputs a single level of cacheability and a user flag (OC) specifying if the architectural attributes were cacheable in the outer cacheable domain. If the transaction is classified as a PCIe one then the NoSnoop transform described above is applied. If the original transaction was NoSnoop, then any weaker memory type is strengthened to iNC-oNC-osh so apply the following transform:

```

if      iWB-oWB-nsh/ish/osh      then output WB-nsh/ish/osh, OC = 1
else if i (NC/WT/WB) -o (WB/WT) then output NC-Sys,      OC = 1
else if i (NC/WT/WB) -oNC        then output NC-Sys,      OC = 0
else if Device-(GRE/nGRE/nGnRE) then output DV-Sys,      OC = 0
else                               output SO-Sys,      OC = 0

```

- The model only normalizes according to PCIe but otherwise leaves the architectural attributes intact on the output bus.
- MSIs are issued on the `qtw_pvbus_m` port using attributes determined by the parameter `msi_attribute_transform`, whilst Event Queue writes are always issued with `ExtendedID=0`, `UserFlags=0`, `ManagerID64=0xffffffff`.

In the hardware, there is no way of distinguishing Event Queue writes from MSI writes, however, this provides a mechanism that if the model system needs to distinguish then it can. MPAM and MEC attributes are provided by the parameters:

- `mpam_attribute_transform`
- `mec_attribute_transform` (not all versions support MEC)
- Note that some older versions of MMU-S3 documentation/RTL referred to the `ns_gbpa_abort_init` and `s_gbpa_abort_init` tie-offs as `sup_ns_gbpa_abort_rst` and `sup_s_gbpa_abort_rst`.
- The model supports architectural features and registers matching `rOp0-00eac0` and `r1p0-00eac0`.

## Security State Determination (SSD)

The model uses the term SSD to mean the security state that the transaction, register, structure, or event belongs to. In the SMMUv3 architecture, the sideband signal `sec_sid` holds this information for the transactions, but uses a different encoding.

Before RME, `sec_sid` was a one-bit signal. With RME, in the hardware, it was expanded to two bits. To retain backwards compatibility in the model, `sec_sid` remains as one-bit in the parameter

howto\_identify, but the second bit can be expressed with SEC\_SID\_bit\_1, and its negative logic version nSEC\_SID\_bit\_1.

Security state	SSD	SEC_SID_bit_1	SEC_SID
secure	0	0	1
non-secure	1	0	0
root (reserved)	2	1	1
realm	3	1	0

Parameters for parsing transaction attributes

Some of the parameters are strings that share a common parsing format for extracting from and placing information into the transaction's attributes.

They can extract and place information into the following fields of a transaction's attributes:

- ManagerID64 (64 bits)
- ExtendedID (64 bits)
- UserFlags (32 bits)

The string parameter is parsed as a comma-separated list of:

```
lhs_expression=rhs_expression
```

The lhs\_expression and rhs\_expression can be an entire symbol, for example:

```
StreamID
```

or a bit, or bit slice:

```
StreamID[16]  
StreamID[31:20]
```

In rhs\_expression, numeric constants (or slices of numeric constants) are also allowed:

```
StreamID[16]=1  
StreamID[16]=10[2]
```

A single symbol might be assigned from multiple non-contiguous arrays of bits from a mix of different RHS symbols:

```
StreamID[16]=1, StreamID[31:30]=ExtendedID[1:0],  
StreamID[15:0]=UserFlags[31:16], ...
```

In those cases where a left hand symbol can also appear on the right hand side, it is possible to swap bits and transform the symbol. For example, the following expression swaps bits 0 and 1 of the ExtendedID:

```
ExtendedID[0]=ExtendedID[1], ExtendedID[1]=ExtendedID[0]
```

Any bits of the attributes that have no transform specified are retained from the input.

The `lhs_expression` and `rhs_expression` must have the same bit width.

### Iris and MTI instances for MMU\_S3

This model has the following Iris instances:

Name	Instance type
MMU_S3	MMU_S3
MMU_S3.axi_stream_msi_manager[4294967295].pvbusmaster	PVBusMaster
MMU_S3.pvbus_id_routed_m[Y].pvbusmaster (where Y = 0,4294967295)	PVBusMaster
MMU_S3.register_file[0]	PVBusSlave
MMU_S3.service_request_tbu[Y] (where Y = 0-63)	PVBusSlave
MMU_S3.tbu[Y] (where Y = 0-63)	PVBusMapper
MMU_S3.tw_msi_qs_manager[4294967295].pvbusmaster	PVBusMaster

This model has the following MTI trace components:

Name	Component type
MMU_S3	MMU_S3
MMU_S3.axi_stream_msi_manager[4294967295].pvbusmaster	PVBusMaster
MMU_S3.pvbus_id_routed_m[Y].pvbusmaster (where Y = 0,4294967295)	PVBusMaster
MMU_S3.register_file[0]	PVBusSlave
MMU_S3.service_request_tbu[Y] (where Y = 0-63)	PVBusSlave
MMU_S3.tbu[Y] (where Y = 0-63)	PVBusMapper
MMU_S3.tw_msi_qs_manager[4294967295].pvbusmaster	PVBusMaster



## Ports for MMU\_S3

Port	Direction	Protocol	Description
axi_stream_msi_addr_to_match_s	slave	Value_64	Any SMMU-originated MSI which exactly matches the address in this port will be sent through the AXI stream port axi_stream_msi_m which is usually connected to the GIC through axi_stream_msi_s. As MSIs are 32 bit aligned then if bits[1:0] != 0 or the address is above OAS then this is effectively disabled. NOTE that the model does not support tcu_sid[31:0] which is the MSI DeviceID to send on axi_stream_msi_m. Instead the parameter smmu_msi_device_id is used. See also the parameters: axi_stream_msi_TID and axi_stream_msi_TDEST. The default value of this port is set by the parameter axi_stream_msi_addr_to_match. This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.
axi_stream_msi_m	master	PVBus	Manager port used for sending SMMU originated MSIs directly to the GIC
clk_in	slave	ClockSignal	Clock signal (in RTL aclk) This is a clock time-base used by the TCU to spread some of its processing over time, if enabled by the wait_* parameters. The clock must always be connected.
cmd_sync_irpt_ns	master	Signal	Pulsed interrupt output signal for non-secure CMD_SYNC having a completion signal of SIG_IRQ.
cmd_sync_irpt_r	master	Signal	Pulsed interrupt output signal for realm CMD_SYNC having a completion signal of SIG_IRQ. This pin exists in r0 but is tied off to 0. It is implemented in r1.
cmd_sync_irpt_s	master	Signal	Pulsed interrupt output signal for secure CMD_SYNC having a completion signal of SIG_IRQ.
event_q_irpt_ns	master	Signal	Pulsed interrupt output signal for the non-secure event queue becoming non-empty.
event_q_irpt_r	master	Signal	Pulsed interrupt output signal for the realm event queue becoming non-empty. This pin exists in r0 but is tied off to 0. It is implemented in r1.
event_q_irpt_s	master	Signal	Pulsed interrupt output signal for the secure event queue becoming non-empty.
evento	master	Signal	Event signal This aligns with the eventoreq signal on the RTL. The eventack signal is not supported.
global_irpt_ns	master	Signal	Pulsed interrupt output signal for non-secure SMMU_GERROR(N) signalling an error.
global_irpt_r	master	Signal	Pulsed interrupt output signal for realm SMMU_R_GERROR(N) signalling an error. This pin exists in r0 but is tied off to 0. It is implemented in r1.
global_irpt_s	master	Signal	Pulsed interrupt output signal for secure SMMU_S_GERROR(N) signalling an error.
gpf_far	master	Signal	An error becomes active in SMMU_ROOT_GPF_FAR.
gpt_cfg_far	master	Signal	An error becomes active in SMMU_ROOT_GPT_CFG_FAR.

Port	Direction	Protocol	Description
identify	master	SMMUv3AEMIdentifyProtocol	Map the transaction to the tuple (StreamID, SubStreamID, SubStreamIDValid, SSD) The StreamID that is produced by the implementation of this protocol is not the final StreamID. The final StreamID is produced by using the list_of_ns_sid_high_at_bitpos0/ list_of_s_sid_high_at_bitpos0 parameter to map the StreamID based on the upstream port index. Also see the parameter howto_identify which can replace the functionality of this port under certain circumstances.
logptsz_s	slave	Value	RME: This is a four bit signal that encodes the region size that a single LOGPT entry covers. The default value of this port is derived from the parameter rme_logpt_entry_covers_log2size_in_bytes which is in a different format to the port. If a valid value is driven then it will be put in the field SMMU_ROOT_GPT_BASE_CFG.LOGPTSZ. The port uses the same encoding as the field. If an invalid value is driven to this port and legacy_tz_en is low then the value is reported in the SMMU_ROOT_GPT_BASE_CFG.LOGPTSZ and then all transactions will fault with a GPT Configuration fault (gpt_cfg_far). This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.
legacy_tz_en	slave	Signal	Tie this high to get non-RME behaviour. On the real hardware, then each of the TCUs and the TBUs have a legacy_tz_en and they must all be driven to the same value. In the model, we only have a single version of this pin. This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.
ns_gbpa_abort_init	slave	Signal	This port is an Non-secure global bypass. The ns_gbpa_abort_init signal sets the reset value of SMMU_GBPA.ABORT: 0 - On reset, do not abort all incoming transactions. 1 - On reset, abort all incoming transactions. This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.
pri_q_irpt_ns	master	Signal	Pulsed interrupt output signal for the non-secure PRI queue becoming non-empty.
pri_q_irpt_r	master	Signal	Pulsed interrupt output signal for the realm PRI queue becoming non-empty. This pin exists in r0 but is tied off to 0. It is implemented in r1.
prog_pvbus_s	slave	PVBus	Register subordinate port (in RTL PROG)
pvbus_id_routed_m	master	PVBus	This is a special "id-routed" port for transmitting ATC invalidates upstream into the PCIe EndPoints, it is not a normal bus. See parameter output_id_routed_transform. The FastModels ATC Invalidate and PRI Response protocol specifies how to route and deal with this port.
qtw_pvbus_m	master	PVBus	Manager port used for Table Walks, MSIs and Queue access. Note that although this is a manager port, it can still receive DVM messages.
s_gbpa_abort_init	slave	Signal	This port is an secure global bypass. The s_gbpa_abort_init signal sets the reset value of SMMU_S_GBPA.ABORT: 0 - On reset, do not abort all incoming transactions. 1 - On reset, abort all incoming transactions. This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.

Port	Direction	Protocol	Description
sec_override	slave	Signal	Allow certain registers to be accessible to non-secure accesses from reset, as described in the TCU_SCR register. This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.
sup_btm	slave	Signal	System supports BTM and will be reflected in the IDR registers. This signal can override the value set by the parameters configuring the IDR registers. If BTM (Broadcast Table Maintenance) is not supported then DVM messages will be ignored. This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.
sup_cohacc	slave	Signal	System supports COHACC and will be reflected in the IDR registers. This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.
sup_httu	slave	Signal	System supports HTTU and will be reflected in the value of SMMU_IDR0.HTTU: - 0b00 (HTTU not supported) if sup_httu is 0 - 0b10 (update of AF and Dirty flags supported) if sup_httu is 1. The default value for this pin is 1. This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.
sup_sev	slave	Signal	System supports SEV and will be reflected in the IDR registers. This signal can override the value set by the parameters configuring the IDR registers. This port is sampled at negedge of tcu_reset_in and must be set before the negedge of the reset signal.
tbm_pvbus_m	master	PVBus	The TBU manager ports that carry transactions that have been translated from the correspondingly numbered tbs_pvbus_s[] port.
tbs_pvbus_s	slave	PVBus	The TBU subordinate ports that receive transactions to be translated. They will exit the SMMU through the same numbered pvbus_m[] port.
tbu_crit_err	master	Signal	Critical error. This cannot occur in the model except by using an Integration Register to generate it.
tbu_pmu_irpt	master	Signal	TBU Performance Monitoring Unit interrupt, one per TBU.
tbu_pmusnapshot_ack	master	Signal	PMU snapshot interface for the TBU, ack a snapshot.
tbu_pmusnapshot_req	slave	Signal	PMU snapshot interface for the TBU, request a snapshot.
tbu_ras_cri	master	Signal	Critical error interrupt for RAS events from the TBU. This is used for specific uncorrected errors where a System Coprocessor (SCP) might have to reset the system because the stability of the system can no longer be guaranteed. NOTE that in the MMU-700 model this is called tbu_cri_irpt.
tbu_ras_eri	master	Signal	Error Recovery Interrupt for RAS events from the TBU. This is used for uncorrected errors. NOTE that in the MMU-700 model this is called tbu_eri_irpt. NOTE that in the MMU-700 model this is called tbu_eri_irpt.
tbu_ras_fhi	master	Signal	Fault Handling Interrupt for RAS events from the TBU. Usually this is for corrected errors. NOTE that in the MMU-700 model this is called tbu_fhi_irpt.
tbu_ras_lt_cri	master	Signal	Level triggered critical error interrupt for RAS events from the TBU.
tbu_ras_lt_eri	master	Signal	Level triggered error recovery interrupt for RAS events from the TBU.

Port	Direction	Protocol	Description
tbu_ras_lt_fhi	master	Signal	Level triggered fault handling interrupt for RAS events from the TBU.
tbu_ras_lt_irpt_v	master	Signal	Level triggered valid output for connection to System RAS agents. Asserted when the RAS record contains at least one valid error.
tbu_reset_in	slave	Signal	Reset signals The TBUs can have independent reset signals. However, TCU reset will result in the reset of all TBUs. This behavior is unlike SMMU RTL implementations which do allow for independent reset of the TCU separate from TBUs. Each signal tbu_reset_in[n] corresponds to the TBU using tbs_pvbus_s[n]/tbs_pvbus_s[n] pair. If the SMMU receives a transaction whilst the TBU is expected to be in reset then it will complain using the ArchMsg.Warning.warning trace source. Those tbu_reset_in that correspond to a PCIe-RC connection can be connected to monitor the PCIe-RC's reset signal. If it receives an ATS request when in reset then it will complain in a similar way. You must connect these pins if you wish the TCU_NODE_STATUS for the nodes to be accurate (including any connected to the PCIe-RC).
tcu_pmu_irpt	master	Signal	TCU Performance Monitoring Unit interrupt
tcu_pmusnapshot_ack	master	Signal	PMU snapshot interface for the TCU, ack a snapshot.
tcu_pmusnapshot_req	slave	Signal	PMU snapshot interface This is a four-phase handshake to have the corresponding PMCG perform a capture of the current counter values. PMU snapshot interface for the TCU, request a snapshot.
tcu_ras_cri	master	Signal	Critical error interrupt for RAS events from the TCU. This is used for specific uncorrected errors where a System Coprocessor (SCP) might have to reset the system because the stability of the system can no longer be guaranteed. NOTE that in the MMU-700 model this is called tcu_cri_irpt.
tcu_ras_eri	master	Signal	Error Recovery Interrupt for RAS events from the TCU. This is used for uncorrected errors. NOTE that in the MMU-700 model this is called tcu_eri_irpt.
tcu_ras_fhi	master	Signal	Fault Handling Interrupt for RAS events from the TCU. Usually this is for corrected errors. NOTE that in the MMU-700 model this is called tcu_fhi_irpt.
tcu_ras_lt_cri	master	Signal	Level triggered critical error interrupt for RAS events from the TCU.
tcu_ras_lt_eri	master	Signal	Level triggered error recovery interrupt for RAS events from the TCU.
tcu_ras_lt_fhi	master	Signal	Level triggered fault handling interrupt for RAS events from the TCU.
tcu_ras_lt_irpt_v	master	Signal	Level triggered valid output for connection to System RAS agents. Asserted when the RAS record contains at least one valid error.
tcu_reset_in	slave	Signal	The reset signal to the TCU interface.

## Parameters for MMU\_S3

### TCUCFG\_DPT\_SUPPORT

MMU-S3 r1 Parameter Only: Enable Device Permission Table (DPT), a mechanism to enforce the association between granules of physical address space and the memory footprint of virtual machines.

**0**

Realm DPT is disabled

**1**

Realm DPT is enabled (default).

Type: `bool`Default value: `true`**TCUCFG\_DVM\_VAS**

Virtual address size used by the system. Once set, this value is discoverable using `TCU_SYSDISC35.TCUCFG_DVM_VAS`.

In hardware, it is important to get this parameter correct as it determines the DVM message format. If this doesn't match the PEs, DVM messages are misinterpreted and any TLBI operations performed are incorrectly applied.

The model uses a representation of DVM that does not depend on the VA size and so misconfiguring this has no effect other than on the system discovery register value. Accepted Values: 49 53.

Type: `uint32_t`

Default value: 53

**TCUCFG\_MECID\_WIDTH**

Memory Encryption Context (MEC) is a feature introduced in MMU-S3. The MECID is a 1-16 bit identifier that, if implemented, supports Memory Encryption Contexts for the Realm programming interface. The given value indicates the number of bits in the MECID. A value of 0 will disable MEC. Accepted Values: 0 4 8 12 16.

Type: `uint32_t`

Default value: 16

**TCUCFG\_PARTID\_WIDTH**

The width of the MPAM PARTID on the bus.

The value 10 is just for MMU\_S3 r1.

See also parameter `mpam_attribute_transform`. Accepted Values: 1 6 9 10.

Type: `unsigned`

Default value: 9

## TCUCFG\_XLATE\_SLOTS

Maximum number of outstanding stalled transactions that the SMMU supports.



TCUCFG\_XLATE\_SLOTS must be  $\geq$  TCUCFG\_PTW\_SLOTS which is currently fixed to 512.

---

Accepted Values: 512 1024 2048 4096.

Type: `uint32_t`

Default value: 512

## all\_error\_messages\_through\_trace

Some conditions in the SMMU are so strange that the software programming the SMMU has done something wrong. At this point messages are output to either `ArchMsg.Error.*` or `ArchMsg.Warning.*` or to the error stream of the simulator. Outputting to the error stream of the simulator may cause it to return with a non-zero exit status.

If you set this option to true then instead of using the error stream of the simulator it will always use a trace stream allowing the simulation to exit with a zero exit status.

Type: `bool`

Default value: `false`

## axi\_stream\_msi\_TDEST

ID of the AXI stream subordinate port on the GIC that receives SMMU originated MSIs sent directly. The name of the GIC port is `axi_stream_msi_s`.



If `axi_stream_msi_addr_to_match[1:0] != 0`, or the address is above OAS then this is effectively disabled.

---

See also:

- `axi_stream_msi_addr_to_match`
- `axi_stream_msi_TID`.

Type: `uint32_t`

Default value: 0

### **axi\_stream\_msi\_TID**

ID of the AXI stream manager port that sends SMMU TCU originated MSIs directly to the GIC. The name of the SMMU port is `axi_stream_msi_m`.



If `axi_stream_msi_addr_to_match[1:0] != 0`, or the address is above OAS then this is effectively disabled.

---

See also:

- `axi_stream_msi_addr_to_match`
- `axi_stream_msi_TDEST`.

Type: `uint32_t`

Default value: 0

### **axi\_stream\_msi\_addr\_to\_match**

If the last two bits are 0, any SMMU-originated MSI which exactly matches the address will be sent through the AXI stream port `axi_stream_msi_m` which is usually connected to the GIC.

This parameter drives the value of the `axi_stream_msi_addr_to_match_s` port at simulation reset. For every reset after that, the value of the port will be sampled and used if changed.



The entire address must match, including bits [1:0]. As MSIs are 32 bit aligned then if `axi_stream_msi_addr_to_match[1:0] != 0`, or the address is above OAS then this is effectively disabled.

---

See also:

- `axi_stream_msi_TID`
- `axi_stream_msi_TDEST`.

Type: `uint64_t`

Default value: `0xFFFFFFFFFFFFFFFF`

### **behaviour\_of\_sampled\_at\_reset\_signals**

Some configuration signals into the SMMU are sampled on negedge of reset.

However, it can sometimes be hard to arrange to drive a configuration pin before the negedge of reset.

The configuration pins are sampled:

**0**

at negedge reset.

**1**

at negedge reset, but if a later change occurs at the same simulated time, and no transactions have occurred, then they will be resampled and the SMMU reset again.

Type: unsigned

Default value: 0

**cmdq\_max\_number\_of\_commands\_to\_buffer**

The command queues can buffer fetched commands before issuing them. This parameter is roughly the maximum number of commands to do this for. The programmer visible effects are that just because the CONS pointer shows a command has been consumed does not necessarily mean that it has been issued (and completed). Higher values accentuate this effect.

Type: uint32\_t

Default value: 10

**enable\_device\_id\_checks**

If this parameter is true then the DeviceIDs seen by the GIC are:

- **for client devices**

$$\text{DeviceID} = \text{StreamID} + \text{translated\_device\_id\_base}$$

- **for SMMU-generated MSIs**

$$\text{smmu\_msi\_device\_id}$$

This parameter enables two checks:

- If the DeviceID is used in the `output_attribute_transform` parameter, if it overflows 32 bits then the model warns. If the DeviceID is not used, it is assumed that the external agent that forms the DeviceID warns if it overflows.
- If the SMMU supports MSIs, the model checks that the GIC is able to distinguish an MSI generated by the SMMU from one generated by a client device.

As the exact mechanism to determine the DeviceID is in the system and not necessarily under control of the SMMU then you can disable these warnings using this parameter.

See also the parameters: `output_attribute_transform` and `msi_attribute_transform`.

Type: bool

Default value: true

**hide\_warning\_ACCESSSEN\_GPCEN\_set\_to\_1\_in\_a\_single\_write**

The architecture recommends against setting `SMMU_ROOT_CR0.ACCESSSEN` and `SMMU_ROOT_CR0.GPCEN` to 1 in the same write if it is possible that a concurrent client device transaction could appear at



the SMMU. This is because there could be a time window where the client device transaction sees effective values `ACCESSEN == 1` and `GPCEN == 0` and so would bypass GPC checking.

If your system cannot generate this possibility then you can set this parameter to true and turn off the warning that software has set both `ACCESSEN` and `GPCEN` to 1 at the same time.

Type: `bool`

Default value: `false`

### **`hide_warning_EOPD_differs_from_what_would_be_cached`**

When this parameter is set to true, warnings that the effective EOPD value differs from what would be cached in the TLB are disabled. False (warnings are showed) by default.

Type: `bool`

Default value: `false`

### **`hide_warning_NoStreamID_transaction_for_unsupported_PAS_or_MPAM_SP`**

When RME is not supported then a NoStreamID transaction with `PAS[1] == 1` or `MPAM_SP[1] == 1` is treated as though `PAS[1] == 0` and `MPAM_SP[1] == 0`. This is usually a system construction error and is not expected to occur.

The SMMU warns when this occurs, but the warning can be hidden by setting this parameter.

Type: `bool`

Default value: `false`

### **`howto_identify`**

If `use-identify` then the SMMU uses the `identify` port to determine the `ssd`, `streamID`, `subStreamID`. Otherwise, this string extracts them from the transaction's attributes.

Examples:

```
SEC_SID=ExtendedID[63], SSV=ExtendedID[62], SubstreamID=ExtendedID[51:32],
StreamID=ExtendedID[31:0]
```

or

```
nSEC_SID=ExtendedID[63], StreamID=ExtendedID[55:24], nSSV=ExtendedID[20],
SubstreamID=ExtendedID[19:0]
StreamID[31:24]=0, StreamID[23:0]=ExtendedID[23:0], SSV=1[0], ...
```



If you are not using the `use-identify` option then the configuration string is parsed according to the rules in the section ‘Parameters for parsing transaction attributes’.

LHS Symbols:

- SIDV**  
Indicates that the StreamID is valid.
- SSV**  
Indicates that the SubstreamID is valid.
- SEC\_SID / SEC\_SID\_bit\_1**  
Bits 0 and 1 respectively of the StreamID Security State
- StreamID**  
(32 b) valid if `SIDV` is 1 or both `SIDV` and `nSIDV` are unused.
- SubstreamID**  
(20 b) is valid if `ssv` is true.
- nSEC\_SID / nSEC\_SID\_bit\_1 / nSSV / nSIDV**  
Negative logic of above parameters. Different attributes are independent and can use negative or positive logic.

RHS Symbols:

**ExtendedID / ManagerID64 / UserFlags**  
Incoming PVBUS transaction attributes

`SEC_SID` is 1b in the model. For RME systems, in hardware, then `SEC_SID` is 2b, in the model `SEC_SID_bit_1` represents bit[1].

`SIDV == 0` indicates a NoStreamID transaction and the SSD is the PAS of the transaction, `SEC_SID` is not used.

Negative and positive logic symbols for the same attribute is an error.

The model uses the term SSD (Security State Determination) to mean which security state the transaction, register, structure, event, etc. belongs to.

In the SMMUV3 Architecture, the side-band signal `SEC_SID` holds the information for the transactions, but uses a different encoding.

Before RME, `SEC_SID` was a one bit signal. With RME, in the hardware, it was expanded to 2b. To retain backwards compatibility in the model, `SEC_SID` remains as 1b in this parameter, but the second bit can be expressed with `SEC_SID_bit_1` (and its negative logic version `nSEC_SID_bit_1`)

Security state	SSD	SEC_SID_bit_1	SEC_SID
secure	0	0	1

Security state	SSD	SEC_SID_bit_1	SEC_SID
non-secure	1	0	0
root (reserved)	2	1	1
realm	3	1	0

For those systems that support NoStreamID transactions, and `howto_identify` is not using the port, first the SMMU determines the value of `SIDV` or `NSIDV` to see if the transaction is a NoStreamID transaction (`SIDV == 0` or `NSIDV == 1`). If so then the rest of the `howto_identify` string is ignored as the information extracted relates only to StreamID transactions. Instead more information is extracted using the parameter `howto_identify_NoStreamID_extra_info`.

In AMBA systems, the equivalent signal to `SIDV` is called either `ARMMUVALID` or `AWMMUVALID`. Collectively they are known as `AxMMUVALID`.

Type: string

Default value: "use-identify"

### **`howto_identify_NoStreamID_extra_info`**

The behavior of this parameter depends on `howto_identify`

- if it equals 'use-identify' then this must be "", otherwise there is an error.
- if it identifies a NoStreamID transaction (`SIDV=0`) then this parameter includes one or more of
  - `MPAM_SP`
  - `MPAM_PARTID`
  - `MPAM_PMG`
  - `MECID`
  - `HWATTR_KIND_0`
- in any other case, this parameter is ignored.

Fields set in this parameter must not overlap the `SIDV/NSIDV` fields in `howto_identify`

Example:

```
MPAM_PMG[7:0]=ExtendedID[62:55], MPAM_PARTID[15:0]=ExtendedID[54:39],
MPAM_SP[1:0]=ExtendedID[38:37], MECID[15:0]=UserFlags[31:16]
HWATTR_KIND_0[3:0]=ExtendedID[42:39]
```



**Note**

The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

Type: string

Default value: ""

### **ish\_is\_osh\_DANGER**

When set, any transaction that would use the architectural inner shareable domain is converted to use the outer shareable domain.



This parameter should match the equivalent `ish_is_osh` from the PE. If an incompatible value of the `ish_is_osh` parameter is configured for the PE and the SMMU, data coherency may be compromised.



All implementations should have this parameter as true but it is allowed in the model to give it a false value for modelling and debug purposes only.

---

Type: `bool`

Default value: `true`

### **legacy\_tz\_en**

The default value of the `legacy_tz_en` pin:

**0**

RME is enabled

**1**

RME is disabled.

Type: `bool`

Default value: `false`

### **list\_of\_ns\_sid\_high\_at\_bitpos0**

A comma-separated list of values to bitwise OR into each Non-secure StreamID for each TBU/Node. Bit 0 of the value corresponds to bit 0 of the StreamID.

Each TBU that is connected to a PCIe-RC (see `list_of_pcie_rc`) must serve a unique contiguous subset of StreamIDs as determined by their top bits. This is used in order to know which port to route ATC Invalidates and PRI Responses to the PCIe subsystems. The effective value for any PCIe-RC ports must be the same for non-secure and realm. See `list_of_r_sid_high_at_bitpos0`.

The empty string corresponds to all 0s.

Type: `string`

Default value: ""

**list\_of\_pcie\_mode**

A comma-separated list of ranges of ports that represent TBUs that are attached to PCIe Root Complexes (PCIe-RC). A single PCIe-RC might use several TBUs and stripe accesses across them. The attribute handling for these TBUs is slightly different in that if the PCIe transaction is NoSnoop and the output attributes of the translation would be weaker than `INC-ONC-osh` then the output is forced to `INC-ONC-osh`.

`INC-ONC-osh` == "inner normal non-cacheable, out normal non-cacheable, outer shared".

Type: `string`

Default value: ""

**list\_of\_pcie\_rc**

This is a list of ports that are connected to PCIe Root Complex (PCIe-RC) by a protocol called DTI-ATS. This port is used to transport ATS and PRI Requests to the SMMU from the PCIe-RC.

In the real hardware, the PCIe-RC uses this port for ATS/PRI, and the actual transactions go through separate TBUs. In the model, this port can accept actual transactions as well. However, in the model, then the ATC Invalidates and the PRI Responses need to be transferred over the corresponding `pvbus_id_routed_m` port as DTI-ATS is bidirectional, but PVBUS is not.

Type: `string`

Default value: ""

**list\_of\_r\_sid\_high\_at\_bitpos0**

A comma-separated list of values to bitwise OR into each Realm StreamID for each TBU/Node. Bit 0 of the value corresponds to bit 0 of the StreamID. This only has an effect for `r1` and later.

Each TBU that is connected to a PCIe-RC (see `list_of_pcie_rc`) must serve a unique contiguous subset of StreamIDs as determined by their top bits. This is used in order to know which port to route ATC Invalidates and PRI Responses to the PCIe subsystems. The effective value for any PCIe-RC ports must be the same for non-secure and realm. See `list_of_ns_sid_high_at_bitpos0`.

The empty string corresponds to all 0s.

"use-ns" can be used to apply the `list_of_ns_sid_high_at_bitpos0` values instead.

Type: `string`

Default value: ""

**list\_of\_s\_sid\_high\_at\_bitpos0**

A comma-separated list of values to bitwise OR into each Secure StreamID for each TBU/Node. Bit 0 of the value corresponds to bit 0 of the StreamID.

The empty string corresponds to all 0s.

use-ns can be used to apply the list\_of\_ns\_sid\_high\_at\_bitpos0 values instead.

Type: string

Default value: ""

### **mec\_attribute\_transform**



The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

If MEC is supported, this is applied to *all* downstream transactions to transport the MEC information.

- "" or "none" – no transform
- How to alter the output attributes. Example:

```
"UserFlags[31:16]=MECID[15:0]"
```

RHS/LHS Symbols:

- ExtendedID/ManagerID64/UserFlags.

RHS Symbols:

- MECID

Any bits with no transform are unchanged.

Attribute transforms applied before this:



- for client transactions output\_attribute\_transform / output\_attribute\_transform\_for\_NoStreamID.
- for table walks tw\_qs\_attribute\_transform.
- for MSIs msi\_attribute\_transform.
- if MPAM is enabled mpam\_attribute\_transform.

Type: string

Default value: ""

## mpam\_attribute\_transform

**Note**

The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

If MPAM is supported, this is applied to *all* downstream transactions to transport the MPAM information.

- "" or "none" – no transform
- How to alter the output attributes. Example:

```
"ExtendedID[62:55]=MPAM_PMG, ExtendedID[54:39]=MPAM_PARTID,
ExtendedID[38]=MPAM_SP[0]"
```

RHS/LHS Symbols:

- ExtendedID/ManagerID64/UserFlags.

RHS Symbols:

- MPAM\_PARTID
- MPAM\_PMG
- MPAM\_NS
- MPAM\_SP
- numeric literals

Any bits with no transform are unchanged.

**Note**

- attribute transforms applied before this:
  - for client transactions output\_attribute\_transform / output\_attribute\_transform\_for\_NoStreamID.
  - for table walks tw\_qs\_attribute\_transform.
  - for MSIs msi\_attribute\_transform.
- mec\_attribute\_transform is applied after this.
- for translated transactions from client devices then MPAM\_NS = ! SEC\_SID.

Type: string

Default value: "ExtendedID[62:55]=MPAM\_PMG, ExtendedID[54:39]=MPAM\_PARTID, ExtendedID[38]=MPAM\_NS"

## msi\_attribute\_transform

Transform downstream attributes of MSI transactions.



The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

- "" or "none" – no transform
- How to alter output attributes of SMMU-generated MSIs. Example:

```
"UserFlags[15:0]=smmu_msi_device_id[31:16],
  ManagerID64[15:0]=smmu_msi_device_id[15:0],
  ExtendedID=0"
```

LHS Symbols:

**ExtendedID / ManagerID64 / UserFlags**

Outgoing PVBUS transaction attributes

RHS Symbols:

**smmu\_msi\_device\_id**

The value stored in the parameter with the same name

**interrupt\_kind**

The selected bit corresponds to the interrupts listed below

**0/1**

EVENTQ s/ns

**2**

PRIQ

**3/4**

CMD\_SYNC s/ns

**5/6**

GERROR s/ns

**7/8**

PMCG s/ns

**9/10/11**

RAS FHI/ERI/CRI

**12/13**

gpf\_far/gpt\_cfg\_far

**14/15/16/17**

Realm EVENTQ/PRIQ/CMDQ/GERROR



**HWATTR\_KIND\_0**

PBHA information

**Numeric Literals**

Any number. Ex: 0x1234

ExtendedID/ManagerID64/UserFlags start with values {0, 0xFFFFffff, 0} respectively.

Any bits with no transform are unchanged.

This transform can be used to determine the DeviceID passed to the GIC to distinguish MSIs generated by the SMMU from those generated by client devices.



See also `output_attribute_transform` and `enable_device_id_checks`.



After 11.25 the `interrupt_kind` field was extended to 5 bits. This is strictly a non-backwards compatible change. However, the original 3 bits were insufficient to express all the interrupt kinds that exist.

Type: `string`

Default value: `"ExtendedID[31:0]=smmu_msi_device_id, ManagerID64[31:0]=0xFFFFffff"`

**normalize\_input\_normal\_non\_iWB\_oWB\_to\_iNC\_oNC\_osh\_DANGER**

When set, use Inner Non Cacheable, Outer Non Cacheable, Outer Shareable for any upstream transaction that would use any of the following attributes:

- Normal Non-cacheable Bufferable
- Normal Non-cacheable Non-bufferable
- Write-through



This parameter should match the equivalent configuration from the PE. If an incompatible value of this parameter is configured for the PE and the SMMU, data coherency may be compromised.

All implementations should have this parameter as true but it is allowed in the model to give it a false value for modelling and debug purposes only.

Type: `bool`

Default value: `true`

**ns\_gbpa\_abort\_init**

The default value of the tie off signal `ns_gbpa_abort_init`.

Type: `bool`

Default value: `false`

**number\_of\_ports**

The number of port pairs that the SMMU has.

Type: `unsigned`

Default value: `1`

**output\_attribute\_transform**

Transform downstream attributes of StreamID transactions.



Note

The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

- "" or "none": no transform
- How to alter output attributes. Example:

```
"ExtendedID[15:0]=DeviceID[15:0], UserFlags[31]=nSSV,
UserFlags[19:0]=SubstreamID,
ManagerID64[10]=ManagerID64[11], ManagerID64[11]=ManagerID64[10]"
```

RHS/LHS Symbols:

**ExtendedID / ManagerID64 / UserFlags**

Incoming/Outgoing PVBUS transaction attributes

RHS Symbols:

**DeviceID**

`StreamID + translated_device_id_base`

**StreamID / SubstreamID / SEC\_SID / SEC\_SID\_bit\_1 / SIDV / SSV**

Architectural information. See parameter `howto_identify` for more information.

**nSEC\_SID / nSEC\_SID\_bit\_1 / nSSV / nSIDV**

Negative logic of above parameters. Different attributes are independent and can use negative or positive logic.

**St1PBHA / St2PBHA**

Page Based Hardware Attributes from leaf descriptors (zero if unused).

**STE\_IMPDEF1**

STE[127:116]

**HWATTR\_KIND\_0**

PBHA information

**Numeric Literals**

Any number. Ex: 0x1234

The streamID has had ns\_sid\_high/s\_sid\_high/r\_sid\_high ORred into it for the appropriate TBU.



- mpam\_attribute\_transform and mec\_attribute\_transform are applied in order after this.
- See also output\_attribute\_transform\_for\_NoStreamID for NoStreamID transactions.

Type: string

Default value: "ExtendedID[31:0]=DeviceID"

**output\_attribute\_transform\_for\_NoStreamID**



The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

Transform downstream attributes of NoStreamID transactions.

- "" or "none": no transform
- How to alter output attributes. Example:

```
"ExtendedID[15:0]=0, UserFlags[31]=1, UserFlags[19:0]=0,
ManagerID64[10]=ManagerID64[11],
ManagerID64[11]=ManagerID64[10] ManagerID64[9:6]=HWATTR_KIND_0"
```

RHS/LHS Symbols: \* ExtendedID/ManagerID64/UserFlags: incoming/outgoing attributes.

RHS Symbols: \* SIDV = 0, nSIDV = 1 (fixed values to indicate NoStreamID) \* PAS \* HWATTR\_KIND\_0

Any bits with no transform are unchanged.



- mpam\_attribute\_transform and mec\_attribute\_transform are applied in order after this.
- see also output\_attribute\_transform for StreamID transactions.

Type: `string`

Default value: `"ExtendedID[31:0]=0, ExtendedID[32]=1"`

### **output\_id\_routed\_transform**



Note

The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

The SMMU generates the following ID-routed transaction on the `pvbus_id_routed_m` bus:

- ATC Invalidate
- PRI Response

This parameter controls how the SMMU should express:

- the StreamID
- the Trusted (T) bit

The value is a comma-separated list of assignments:

```
Address[27:12]=StreamID[15:0], ExtendedID[60]=T, ExtendedID[15:0]=StreamID[31:16]
```

Address bits[11:0] cannot be used.

The LHS can be one of:

- PAS
- ManagerID64 / ExtendedID / UserFlags
- Address

The RHS can be one of:

- a numeric constant
- SSD
- T or negative version nT
- StreamID

For realm (or 'Trusted') transactions, then `ssd=0b11`, `T=1`, `nT=0`. For non-secure (or 'Non-Trusted') transactions, then `ssd=0b01`, `T=0`, `nT=1`.

Type: `string`

Default value: `"Address[27:12]=StreamID[15:0], PAS=SSD"`

**prefetch\_only\_requests**

The simulator supports 'prefetch-only' DMI requests, which can occur at any time and for any reason and are intended to be invisible to the end execution of the model and to the user.

**0**

deny all prefetch-only requests

**1**

- use debug requests for any page table walks
  - form and use debug TLB/cache entries
  - any faults will not record, but deny the prefetch request

**2**

- treat prefetch-only requests like normal transactions
  - use normal page table walk transactions
  - use and form normal TLB/cache entries
  - faults will alter the programmer-visible state of the SMMU

0 is the safest.

1 treats the access like a debug request and requires that debug page table walks are treated correctly downstream. Any descriptors that need HTTU to allow the transaction to proceed will fail the request.

2 is dangerous, it uses real transactions and reports faults that are unphysical. Real transactions can be `wait()`ed and this disobeys the SystemC spec for `get_direct_mem_ptr()`.

Type: unsigned

Default value: 0

**rme\_logpt\_entry\_covers\_log2size\_in\_bytes**

Each LOGPT entry covers:

```
2**rme_logpt_entry_covers_log2size_in_bytes
```

bytes of address space.

The valid values for this parameter are: 30, 34, 36, 39

This parameter is reported in an encoded format as the read-only field:

```
SMMU_ROOT_GPT_BASE_CFG.LOGPTSZ
```

This parameter can be overridden by the port `logptsz_s` when sampled on negedge of reset.

Type: `uint32_t`

Default value: 30

### **`s_gbpa_abort_init`**

The default value of the tie off signal `s_gbpa_abort_init`.

Type: `bool`

Default value: `false`

### **`sec_override`**

The IMP DEF port `sec_override` controls whether some of the registers are accessible to secure or non-secure/realms transactions. This parameter is the default value assumed for that port if the port is not driven by a signal.

Type: `bool`

Default value: `false`

### **`seed`**

Used to seed the pseudo-random number generator that the SMMU model uses.

Type: `uint32_t`

Default value: 0x12345678

### **`size_of_cd_cache`**

The number of entries in the cache holding CD structures. If this is zero it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

Type: `uint32_t`

Default value: 0

### **`size_of_dpttlb`**

The number of entries in the DPT TLB. If this is zero then it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

Type: `uint32_t`

Default value: 0

### **`size_of_gpttlb`**

The number of entries in the GPT TLB. If this is zero then it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

Type: `uint32_t`

Default value: 0

### **size\_of\_l1cd\_cache**

The number of entries in the cache holding L1CD descriptors. If this is zero it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

Type: `uint32_t`

Default value: 0

### **size\_of\_l1ste\_cache**

The number of entries in the cache holding L1STE descriptors. If this is zero it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

Type: `uint32_t`

Default value: 0

### **size\_of\_ste\_cache**

The number of entries in the cache holding STE structures. If this is zero it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

Type: `uint32_t`

Default value: 0

### **size\_of\_tlb**

The number of entries in the TLB. If this is zero it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

Type: `uint32_t`

Default value: 0

### **smmu\_msi\_device\_id**

When appropriately enabled, assume that MSIs that are generated by the SMMU are presented to the GIC with this DeviceID.

See parameters `msi_attribute_transform` and `enable_device_id_checks`.

Type: `uint32_t`

Default value: 0

### **sup\_btm**

The default value of the register field `SMMU_IDR0.BTM` and `SMMU_ROOT_IDR0.BGPTM` (when supported). This enables Broadcast TLB maintenance.

Type: `bool`

Default value: `true`

### **`sup_cohacc`**

The default value of the register `SMMU_IDR0.COACC`.

Type: `bool`

Default value: `true`

### **`sup_httu`**

The initial value of the `sup_httu` port. See the port description for `sup_httu`.

Type: `bool`

Default value: `true`

### **`sup_oas`**

The hardware has an input port `sup_oas[2:0]` that indicates what output address size (OAS) the system has. This is sampled at reset.

The model does not have this port as it is expected to be a constant for the system and not to change. Instead it is just a parameter.

The allowed values are:

- |          |          |
|----------|----------|
| <b>0</b> | 32 bits  |
| <b>1</b> | 36 bits  |
| <b>2</b> | 40 bits  |
| <b>3</b> | 42 bits  |
| <b>4</b> | 44 bits  |
| <b>5</b> | 48 bits  |
| <b>6</b> | 52 bits. |

Type: `unsigned`

Default value: 6



**sup\_sev**

The default value of the register `SMMU_IDR0.SEV`.

Type: `bool`

Default value: `true`

**tlb\_when\_do\_f\_tlb\_conflict\_on\_overlap**

If a TLB entry is created by a walk and it overlaps an existing entry, there are some architectural situations where the result is known. For all others, then an implementation is allowed to use an **UNPREDICTABLE** combination of the two entries, or it can generate `F_TLB_CONFLICT`:

**0**

never generate

**1**

sometimes generate

**2**

always generate

Conflicts between global and non-global entries are not detected by the model.

Type: `unsigned`

Default value: `0`

**translated\_device\_id\_base**

When appropriately enabled, assume that client device accesses are translated to a DeviceID as seen by the GIC of:

```
StreamID + translated_device_id_base
```

See parameter `output_attribute_transform` and `enable_device_id_checks`.

Type: `uint32_t`

Default value: `0`

**tw\_qs\_attribute\_transform**

Transform downstream attributes of table walk and queue transactions.



The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

- 
- "" or "none" – no transform

- How to alter the output attributes. Example:

```
"ExtendedID[35:32]=HWATTR_KIND_0"
```

RHS/LHS Symbols:

**ExtendedID / ManagerID64 / UserFlags**

Incoming/Outgoing PVBUS transaction attributes

RHS Symbols:

**HWATTR\_KIND\_0**

PBHA information

**kind**

Transaction kind

- For a read:

**0/1**

L1STE/STE

**2/3**

L1CD/CD

**4/5**

S1/S2 TTD (including CAS)

**6**

CMDQ

**7**

VMS

**11/12**

LOGPT/L1GPT

**13/14**

LODPT/L1DPT

- For a write:

**0**

EVENTQ

**1**

PRIQ

ExtendedID/ManagerID64/UserFlags start with values {0, 0xFFFFFFFF, 0} respectively.

Any bits with no transform are unchanged.



See also `output_attribute_transform` and `msi_attribute_transform`.

---

Type: `string`

Default value: `""`

### **version**

The version of this product.

Valid values are:

- `r0p0`
- `r1p0`.

Type: `string`

Default value: `"r0p0"`

### **wait\_cmdq\_ticks**

This is the time to wait before doing something on the command queue. If bit [32] is set `0x1_0000_0000` then the time waited for is a uniform randomly-distributed time  $[0, \max(2, (t \& 0xFFFFffffff)) - 1]$ .

Type: `uint64_t`

Default value: `0`

### **wait\_eventq\_ticks**

This is the time to wait before doing something on the event queue. If bit [32] is set `0x1_0000_0000` then the time waited for is a uniform randomly-distributed time  $[0, \max(2, (t \& 0xFFFFffffff)) - 1]$ .

Type: `uint64_t`

Default value: `0`

### **wait\_misc\_async\_actions\_ticks**

This is the time to wait before doing an async action that could be delayed is performed. If bit [32] is set `0x1_0000_0000` then the time waited for is a uniform randomly-distributed time  $[0, \max(2, (t \& 0xFFFFffffff)) - 1]$ .

Type: `uint64_t`

Default value: `0`

**wait\_msi\_ticks**

This is the time to wait before sending an MSI. If bit [32] is set 0x1\_0000\_0000 then the time waited for is a uniform randomly-distributed time  $[0, \max(2, (t \ \& \ 0xFFFFFFFF)) - 1]$ .

Type: uint64\_t

Default value: 0

**wait\_pri\_req\_ticks**

This is the time to wait before processing a PRI Request. If bit [32] is set 0x1\_0000\_0000 then the time waited for is a uniform randomly-distributed time  $[0, \max(2, (t \ \& \ 0xFFFFFFFF)) - 1]$ .

Type: uint64\_t

Default value: 0

**wait\_pri\_resp\_ticks**

This is the time to wait before sending a PRI Response back to the PCIe subsystem. When a PRI Response is an auto-response then the ATC might immediately make a new ATS request, that immediately fails, immediately makes a PRI Request, or auto-responds, etc. To break this loop, then we introduce a minimum time on all PRI Responses to give other components in the system a chance to run. If bit [32] is set 0x1\_0000\_0000 then the time waited for is a uniform randomly-distributed time  $[0, \max(2, (t \ \& \ 0xFFFFFFFF)) - 1]$ .

Type: uint64\_t

Default value: 1

## 3.248 MSIRewriter

Defined in `LISA/MSIRewriter.lisa`.

### About MSIRewriter

MSIRewriter is a component that implements the functionality of the MSI-64 Encapsulator in GIC IP, for example GIC-700. For more information about GIC-700, see [GIC-700 Technical Reference Manual](#). For the GIC architecture specification version 3 and version 4, see [GIC Architecture Specification](#).

If an MSIRewriter component is used, it converts writes to the `GITS_TRANSLATER` register to writes to a model-only register called `GITS_TRANSLATE64R`. `GITS_TRANSLATE64R` holds the DeviceID in the upper 32 bits and the EventID in the lower 32 bits. The lower 32 bits of `GITS_TRANSLATE64R` correspond to the `GITS_TRANSLATER` register.

The `GITS_TRANSLATER` register is in a page by itself except for the `GITS_TRANSLATE64R` register.

Any 16/32 bit, single beat write to `GITS_TRANSLATER` is rewritten to a 64 bit write to `GITS_TRANSLATE64R` where the top 32 bits represent the DeviceID and are assumed to come from the bottom 32 bits of ExtendedID.

### ManagerID, ExtendedID, and UserFlags

These tables show how this model encodes information in the `ManagerID`, `ExtendedID`, and `UserFlags` bus attributes:

**Table 3-867: pvbus\_s port**

PVBus attribute	Bits used	Property encoded	Notes
ManagerID	–	–	Not used
ExtendedID	Bits[63:32]	Stream ID	–
	Bits[31:0]	Device ID	–
UserFlags	–	–	Not used

**Table 3-868: pvbus\_m port**

PVBus attribute	Bits used	Property encoded	Notes
ManagerID	Bits[31:16]	Optional 16-bit label	For usage, see the <code>label</code> parameter
ExtendedID	–	–	Not used
UserFlags	–	–	Not used

### Iris and MTI instances for MSIRewriter

This model has the following Iris instances:

Name	Instance type
<code>MSIRewriter</code>	<a href="#">MSIRewriter</a>
<code>MSIRewriter.mapper</code>	<a href="#">PVBusMapper</a>
<code>MSIRewriter.pvbusmaster</code>	<a href="#">PVBusMaster</a>
<code>MSIRewriter.pvbusslave</code>	<a href="#">PVBusSlave</a>

This model has the following MTI trace components:

Name	Component type
<code>MSIRewriter.mapper</code>	<a href="#">PVBusMapper</a>
<code>MSIRewriter.pvbusmaster</code>	<a href="#">PVBusMaster</a>
<code>MSIRewriter.pvbusslave</code>	<a href="#">PVBusSlave</a>

### Ports for MSIRewriter

Port	Direction	Protocol	Description
<code>pvbus_m</code>	master	<a href="#">PVBus</a>	–
<code>pvbus_s</code>	slave	<a href="#">PVBus</a>	–

## Parameters for MSIRewriter

### **GITS\_TRANSLATE64R\_OFFSET**

Offset of GITS\_TRANSLATE64R from the ITS base address. When `rewrite_offset` is non-zero, this option is ignored.

Type: `uint64_t`

Default value: `0x10048`

### **ITS0-base**

Register base address for ITS0. This base address is used to recognise writes to the GITS\_TRANSLATER register within the ITS0's register frame. Ignored when `translate_frame_base_addresses` is non-empty `std::string`.

Type: `uint64_t`

Default value: `N/A`

### **enable\_rewriting**

Enable rewriting.

Type: `bool`

Default value: `true`

### **label**

If  $< 2^{16}$  then this is a label that is put in the [31:16] bits of ManagerID in the same way that the component Labeller does. This labelling is not controlled by `enable_rewriting` and is performed on all transactions (even rewritten ones).

Type: `uint32_t`

Default value: `0xFFFFFFFF`

### **log**

Log level, 0 is off.

Type: `unsigned`

Default value: `0`

### **rewrite\_offset**

The offset of the address where the rewritten transaction should target calculated from the original target address. When this option is non-zero, `GITS_TRANSLATE64R_OFFSET` is ignored.

Type: `int64_t`

Default value: 0

### **translate\_frame\_base\_addresses**

Comma separated list of base addresses for the 64KB translation frames of ITS instances (not the ITS base address). These addresses are used to recognise writes to the GITS\_TRANSLATER register.

Type: `string`

Default value: ""

## 3.249 Mali\_C10

Defined in `LISA/Mali_C10.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p1	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Limitations

- It does not expose any targets for Iris-enabled debuggers to connect to, so debug access to its registers through Iris is not supported.

### Iris and MTI instances for Mali\_C10

This model has the following Iris instances:

Name	Instance type
Mali_C10	Mali_C10
Mali_C10.ExportTest.Mali_C10[0].pvbusmaster	PVBusMaster
Mali_C10.apb_slave	PVBusSlave

This model has the following MTI trace components:

Name	Component type
Mali_C10	Mali_C10
Mali_C10.ExportTest.Mali_C10[0].pvbusmaster	PVBusMaster
Mali_C10.apb_slave	PVBusSlave

### Ports for Mali\_C10

Port	Direction	Protocol	Description
irq	master	Signal	Shared interrupt

Port	Direction	Protocol	Description
pvbus_m	master	PVBus	Master AXI port for RAM access
pvbus_s	slave	PVBus	Slave port for register access
reset_s	slave	Signal	Reset signal

## Parameters for Mali\_C10

### verbosity

Messages verbosity level: -1 - print ERROR messages only, 0 - print ERROR and WARNING messages, 1 - print INFO messages, 2 - print DEBUG messages.

Type: int

Default value: 0

## 3.250 Mali\_C55

Defined in `LISA/Mali_C55.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Preliminary support

For an explanation of the quality levels, see [Quality level definitions](#).

### Limitations

- Streaming input and output interfaces are supported but the protocol differs slightly from the real hardware. See `Mali_Cxx_streaming_camera.lisa` for details on how to use them.
- Video monitor output and DMA monitor interface are not supported.
- Error conditions (IRQ bits 2, 3, 19, 20, 22) are not supported.
- It does not expose any targets for Iris-enabled debuggers to connect to, so debug access to its registers through Iris is not supported.

### Iris and MTI instances for Mali\_C55

This model has the following Iris instances:

Name	Instance type
Mali_C55	Mali_C55
Mali_C55.ExportTest.Mali_C55[0].pvbusmaster	PVBusMaster
Mali_C55.apb_slave	PVBusSlave

This model has the following MTI trace components:



Name	Component type
Mali_C55	Mali_C55
Mali_C55.ExportTest.Mali_C55[0].pvbusmaster	PVBusMaster
Mali_C55.apb_slave	PVBusSlave

## Ports for Mali\_C55

Port	Direction	Protocol	Description
ds_data_out_m	master	PVBus	-
ds_hsync_out_m	master	Signal	-
ds_uv_valid_out_m	master	Signal	-
ds_vsync_out_m	master	Signal	-
fr_data_out_m	master	PVBus	Metadata + pixels, 16 bit, (RGB or YUV)
fr_hsync_out_m	master	Signal	Horizontal sync / blanking (set at the beginning of a line; cleared at the end)
fr_uv_valid_out_m	master	Signal	UV valid (set if both U and V pixels are valid)
fr_vsync_out_m	master	Signal	Vertical sync / blanking (set at the beginning of a frame; cleared at the end)
irq	master	Signal	Shared interrupt
pvbus_m	master	PVBus	Master AXI port for RAM access
pvbus_s	slave	PVBus	Slave port for register access
reset_s	slave	Signal	Reset signal
stream_data_in_s	slave	PVBus	Metadata + pixels, 20 bit
stream_hsync_in_s	slave	Signal	Horizontal sync / blanking (set at the beginning of a line; cleared at the end)
stream_vsync_in_s	slave	Signal	Vertical sync / blanking (set at the beginning of a frame; cleared at the end)

## Parameters for Mali\_C55

### **CNR\_FITTED**

Color Noise Reduction (CNR) and the square and square root for CNR: 0 - absent, 1 - present.

Type: int

Default value: 1

### **COMPRESSION\_FITTED**

Temper compression logic: 0 - absent, 1 - present.

Type: int

Default value: 1

### **DSPIPE\_FITTED**

Downscaled pipeline branch: 0 - absent, 1 - present.

Type: int

Default value: 1

**FRSCALER\_FITTED**

Rull Resolution pipeline RGB scaler: 0 - absent, 1 - present.

Type: `int`

Default value: 1

**IRIDIX\_GTM\_FITTED**

Iridix(TM) global tone-mapping logic: 0 - absent, 1 - present. Must be 1 if IRIDIX\_LTM\_FITTED == 0.

Type: `int`

Default value: 0

**IRIDIX\_LTM\_FITTED**

Iridix(TM) local tone-mapping logic: 0 - absent, 1 - present.

Type: `int`

Default value: 1

**PONG\_CONFIG\_FITTED**

Pong configuration space: 0 - absent, 1 - present.

Type: `int`

Default value: 1

**SCALER\_COEF\_SETS**

Number of scaler coefficient sets (8 or 16).

Type: `int`

Default value: 8

**SINTER\_FITTED**

Sinter block: 0 - absent, 1 - present.

Type: `int`

Default value: 1

**SINTER\_LITE**

Sinter version: 0 - full, 1 - lite.

Type: `int`

Default value: 0

**TEMPER\_FITTED**

Temper, DMA, or merge: 0 - absent, 1 - present.

Type: `int`

Default value: 1

**WDR\_FITTED**

Wide Dynamic Range (WDR) frame stitch, offset, and gain: 0 - absent, 1 - present.

Type: `int`

Default value: 1

**ext\_mode**

Reserved for future use. Use it with instructions from Arm Technical Support (support-esl@arm.com).

Type: `int`

Default value: 0

**verbosity**

Messages verbosity level: -1 - print ERROR messages only, 0 - print ERROR and WARNING messages, 1 - print INFO messages, 2 - print DEBUG messages.

Type: `int`

Default value: 0

3.251 Mali\_C71

Defined in `LISA/Mali_C71.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r1p2	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

## Limitations

- Streaming input and output interfaces are supported but the protocol differs slightly from the real hardware.
- Fault Interface is not supported (except double interrupt).
- It does not expose any targets for Iris-enabled debuggers to connect to, so debug access to its registers through Iris is not supported.

## Iris and MTI instances for Mali\_C71

This model has the following Iris instances:

Name	Instance type
Mali_C71	Mali_C71
Mali_C71.ExportTest.Mali_C71[0].pvbusmaster	PVBusMaster
Mali_C71.apb_slave	PVBusSlave

This model has the following MTI trace components:

Name	Component type
Mali_C71	Mali_C71
Mali_C71.ExportTest.Mali_C71[0].pvbusmaster	PVBusMaster
Mali_C71.apb_slave	PVBusSlave

## Ports for Mali\_C71

Port	Direction	Protocol	Description
fault	master	Signal	Fault output interface
irq	master	Signal	Shared interrupts
pvbus_m	master	PVBus	Master AXI port for RAM access
pvbus_s	slave	PVBus	Slave port for register access
reset_s	slave	Signal	Reset signal.
stream_data_in_s	slave	PVBus	Metadata + pixels
stream_data_out_m	master	PVBus	Metadata + pixels
stream_hsync_in_s	slave	Signal	Horizontal sync / blanking (set at the beginning of a line; cleared at the end)
stream_hsync_out_m	master	Signal	Horizontal sync / blanking (set at the beginning of a line; cleared at the end)
stream_tag_in_s	slave	Value	Source stream tag Can change between lines (while hsync is Clear) if several frames are multiplexed.
stream_tag_out_m	master	Value	Source stream tag (changes while vsync is Clear)
stream_vsync_in_s	slave	Signal	Vertical sync / blanking (set at the beginning of a frame; cleared at the end)
stream_vsync_out_m	master	Signal	Vertical sync / blanking (set at the beginning of a frame; cleared at the end)

## Parameters for Mali\_C71

### **verbosity**

Messages verbosity level: -1 - print ERROR messages only, 0 - print ERROR and WARNING messages, 1 - print INFO messages, 2 - print DEBUG messages.

Type: `int`

Default value: 0

## 3.252 Mali\_C720AE

Defined in `LISA/Mali_C720AE.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Preliminary support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

Model quality level changes:

From	To
Alpha support	Preliminary support

### Limitations

- Streaming input and output interfaces are supported, but the protocol differs slightly from the actual hardware.
- The interrupt interface has limited support.
- The fault interface is not supported.
- This model does not expose any targets for Iris-enabled debuggers to connect to, so debug access to its registers through Iris is not supported.

### Iris and MTI instances for Mali\_C720AE

This model has the following Iris instances:

Name	Instance type
<code>Mali_C720AE</code>	<code>Mali_C720AE</code>
<code>Mali_C720AE.ExportTest.Mali_C720AE[0].pvbusmaster</code>	<code>PVBusMaster</code>
<code>Mali_C720AE.apb_slave</code>	<code>PVBusSlave</code>

This model has the following MTI trace components:

Name	Component type
Mali_C720AE	Mali_C720AE
Mali_C720AE.ExportTest.Mali_C720AE[0].pvbusmaster	PVBusMaster
Mali_C720AE.apb_slave	PVBusSlave

### Ports for Mali\_C720AE

Port	Direction	Protocol	Description
fault	master	Signal	Fault output interface
irq	master	Signal	Shared interrupt
pvbus_m	master	PVBus	Master AXI port for RAM access
pvbus_s	slave	PVBus	Slave port for register access
reset_s	slave	Signal	Reset signal.
stream_data_in_s	slave	PVBus	Metadata + pixels
stream_data_out_m	master	PVBus	Metadata + pixels
stream_hsync_in_s	slave	Signal	Horizontal sync / blanking (set at the beginning of a line; cleared at the end)
stream_hsync_out_m	master	Signal	Horizontal sync / blanking (set at the beginning of a line; cleared at the end)
stream_tag_in_s	slave	Value	Source stream tag Can change between lines (while hsync is Clear) if several frames are multiplexed.
stream_tag_out_m	master	Value	Source stream tag (changes while vsync is Clear)
stream_vsync_in_s	slave	Signal	Vertical sync / blanking (set at the beginning of a frame; cleared at the end)
stream_vsync_out_m	master	Signal	Vertical sync / blanking (set at the beginning of a frame; cleared at the end)

### Parameters for Mali\_C720AE

#### verbosity

Messages verbosity level: -1 - print ERROR messages only, 0 - print ERROR and WARNING messages, 1 - print INFO messages, 2 - print DEBUG messages.

Type: int

Default value: 0

## 3.253 Mali\_C78

Defined in LISA/Mali\_C78.lisa.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r1p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

## Limitations

- Streaming input and output interfaces are supported but the protocol differs slightly from the real hardware.
- Fault Interface is not supported (except double interrupt).
- It does not expose any targets for Iris-enabled debuggers to connect to, so debug access to its registers through Iris is not supported.

## Iris and MTI instances for Mali\_C78

This model has the following Iris instances:

Name	Instance type
Mali_C78	Mali_C78
Mali_C78.ExportTest.Mali_C78[0].pvbusmaster	PVBusMaster
Mali_C78.apb_slave	PVBusSlave

This model has the following MTI trace components:

Name	Component type
Mali_C78	Mali_C78
Mali_C78.ExportTest.Mali_C78[0].pvbusmaster	PVBusMaster
Mali_C78.apb_slave	PVBusSlave

## Ports for Mali\_C78

Port	Direction	Protocol	Description
fault	master	Signal	Fault output interface
irq	master	Signal	Shared interrupts
pvbus_m	master	PVBus	Master AXI port for RAM access
pvbus_s	slave	PVBus	Slave port for register access
reset_s	slave	Signal	Reset signal.
stream_data_in_s	slave	PVBus	Metadata + pixels
stream_data_out_m	master	PVBus	Metadata + pixels
stream_hsync_in_s	slave	Signal	Horizontal sync / blanking (set at the beginning of a line; cleared at the end)
stream_hsync_out_m	master	Signal	Horizontal sync / blanking (set at the beginning of a line; cleared at the end)
stream_tag_in_s	slave	Value	Source stream tag Can change between lines (while hsync is Clear) if several frames are multiplexed.
stream_tag_out_m	master	Value	Source stream tag (changes while vsync is Clear)
stream_vsync_in_s	slave	Signal	Vertical sync / blanking (set at the beginning of a frame; cleared at the end)
stream_vsync_out_m	master	Signal	Vertical sync / blanking (set at the beginning of a frame; cleared at the end)

## Parameters for Mali\_C78

### **verbosity**

Messages verbosity level: -1 - print ERROR messages only, 0 - print ERROR and WARNING messages, 1 - print INFO messages, 2 - print DEBUG messages.

Type: `int`

Default value: 0

## 3.254 Mali\_Cxx\_streaming\_camera

Defined in `LISA/Mali_Cxx_streaming_camera.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### **About Mali\_Cxx\_streaming\_camera**

Simple streaming camera to connect to Mali ISP (C55, C71, or C78) streaming inputs. This component tests ISP's streaming input functionality and provides an example of how to develop camera-like components streaming frames to a Mali-Cxx ISP. All the implementation code is placed in this LISA+ file.

### **Iris and MTI instances for Mali\_Cxx\_streaming\_camera**

This model has the following Iris instances:

Name	Instance type
<code>Mali_Cxx_streaming_camera</code>	<a href="#">Mali_Cxx_streaming_camera</a>
<code>Mali_Cxx_streaming_camera.bus_master_data</code>	<a href="#">PVBUSMaster</a>
<code>Mali_Cxx_streaming_camera.bus_slave_config</code>	<a href="#">PVBUSSlave</a>

This model has the following MTI trace components:

Name	Component type
<code>Mali_Cxx_streaming_camera.bus_master_data</code>	<a href="#">PVBUSMaster</a>
<code>Mali_Cxx_streaming_camera.bus_slave_config</code>	<a href="#">PVBUSSlave</a>



## Ports for Mali\_Cxx\_streaming\_camera

Port	Direction	Protocol	Description
config_s	slave	PVBus	Access to the camera config register(s) (8 bit): 0x00 - output mode: 0 - no output, 1 - stream one frame and auto-reset to 0
data_m	master	PVBus	Output frames port; sends pixels as 32-bit values (and debug metadata).
hsync_m	master	Signal	Horizontal sync; set at the beginning of a line, cleared at the end.
vsync_m	master	Signal	Vertical sync; set at the beginning of a frame, cleared at the end.

## Parameters for Mali\_Cxx\_streaming\_camera

### image\_file

A file containing one or more frames to stream in the ISP model's FRM format.

Type: `string`

Default value: `N/A`

### output\_bpp

Bits per pixel of output image (MSB aligned). 0 means use of input FRM file value.

Type: `int`

Default value: `20`

## 3.255 Mali\_Cxx\_streaming\_sink

Defined in `LISA/Mali_Cxx_streaming_sink.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### About Mali\_Cxx\_streaming\_sink

Arm Mali ISP capture component example. This component tests ISP's streaming output functionality and provides an example of how to develop components consuming frames from the Mali ISP streaming output. This component can be attached as a consumer to either one plane or 3 planes Mali ISP streaming output port and will capture (save) all the received frames to the host disk. Planes are saved to separate files. This implementation saves all the ISP output frames in the FRM file format which is defined as part of the ISP model specification. All the implementation source code is placed in this LISA+ file.

## Iris and MTI instances for Mali\_Cxx\_streaming\_sink

This model has the following Iris instances:

Name	Instance type
Mali_Cxx_streaming_sink	Mali_Cxx_streaming_sink
Mali_Cxx_streaming_sink.bus_slaveX (where X = 0-2)	PVBusSlave

This model has the following MTI trace components:

Name	Component type
Mali_Cxx_streaming_sink.bus_slaveX (where X = 0-2)	PVBusSlave

## Ports for Mali\_Cxx\_streaming\_sink

Port	Direction	Protocol	Description
data_s	slave	PVBus	Pixels consumer ports; receive pixels as 8-bit values (and optional debug metadata).
hsync_s	slave	Signal	Horizontal sync port; defines input image's line begin and end for all three data ports.
vsync_s	slave	Signal	Vertical sync port; defines the input frame's begin and end for all three data ports.

## Parameters for Mali\_Cxx\_streaming\_sink

### do\_capture

Saving captured frames on/off.

Type: bool

Default value: true

### fn\_prefix

Saved filenames prefix.

Type: string

Default value: "isp\_out\_"

### pixel\_bytes

Bytes per pixel in input data.

Type: int

Default value: 2

### planes\_number

Color planes number.

Type: int

Default value: 1

## 3.256 Mali\_G1

Defined in `LISA/Mali_G1.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0 r0p1	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### About Mali\_G1

The Mali\_G1 model is part of a new generation of fully-functional GPU models that generate correct output without the assistance of the Generic Graphics Accelerator (GGA) add-on.

The files related to Mali\_G1 included in the package are:

- `$(PVLIB_HOME)/LISA/Mali_G1.lisa`
- `$(PVLIB_HOME)/etc/g1model.sgrep0`
- `$(PVLIB_HOME)/lib/<platform-compiler>/libMaliG1Model.so`

To keep CPU and GPU performance aligned, Mali\_G1 requires a clock input running at around 500 MHz. This avoids software timeouts during slow-running GPU operations.

The following testing has been performed:

- Target OS:
  - Debian Bookworm running Linux 6.6
- Mali DDK:
  - r54p0-00eac0

### Operating modes

The model supports multiple operating modes which trade performance against functional accuracy. The operating mode is selected using the `mode` parameter.

#### Fast mode

The default and recommended operating mode. It provides the highest level of functional accuracy across graphics and compute workloads and is the most robust option for general use, validation, and software bring-up.

#### Turbo mode

Enables a just-in-time execution mechanism which significantly improves performance, particularly for compute-focused workloads. This mode is still under active development and is not yet a fully general-purpose execution mode. Compute workloads are expected to

work correctly in turbo mode, and generally see the largest performance benefit. Graphics workloads may function in simple cases, but coverage is incomplete and failures are possible.

Turbo mode has known limitations and is not fully validated across all APIs. For workloads requiring stability or broad graphics support, fast mode is recommended.

## Limitations

- The model does not support debug access to registers through Iris-enabled debuggers.
- The timing data observed from the model does not correspond in any way to real hardware.
- The model is not supported on Windows hosts.
- Not all performance counters are updated, and those that are may not always report accurate values.
- The model may not cleanly recover from GPU reset, suspend, or resume operations. To work around this, it is recommended that any GPU governors are left in an always-on mode, and the operating system is configured to avoid GPU resets due to lack of progress. Driver timeout settings may also need adjustment to accomodate slower processing.
- Neural Accelerator and hardware ray-tracing pipelines are modelled but have limited validation coverage and should be considered preview quality.

## Iris and MTI instances for Mali\_G1

This model has the following Iris instances:

Name	Instance type
Mali_G1	Mali_G1
Mali_G1.ExportTest.Mali_G1[0].pvbusmaster	PVBusMaster
Mali_G1.busmaster	PVBusMaster
Mali_G1.subordinate	PVBusSlave

This model has the following MTI trace components:

Name	Component type
Mali_G1	Mali_G1
Mali_G1.ExportTest.Mali_G1[0].pvbusmaster	PVBusMaster
Mali_G1.busmaster	PVBusMaster
Mali_G1.subordinate	PVBusSlave

## Ports for Mali\_G1

Port	Direction	Protocol	Description
clk_in	slave	ClockSignal	-
cntvalueb	slave	CounterInterface	Interface to the SoC-level Generic Counter module that is used to source the global system timestamp value. When the GPU component is integrated into an A-Profile compliant system, this port must be connected for the GPU and CPU timestamps to be synchronized. Without the connection, the GPU timestamp remains constant. In this case, it is recommended to configure the GPU to use its own internal clock as the reference.

Port	Direction	Protocol	Description
gpu_reset	slave	Signal	Resets the GPU at the input Set. The available inputs are Set and Clear.
irq_aw	master	Signal	The interrupt signal generated from one of the GPU Access Windows.
prot_mode_m0	master	Signal	Indicates the current state of the GPU. When it outputs: Set: The GPU is in protected mode. Clear: The GPU is in normal mode. Typically, connect this signal to the LabellerForGPUProtMode component. This enables the labeller to add a Non-Secure Access ID to the outgoing transaction on the labeller's pvbus_m port.
pvbus_m	master	PVBus	The interface for the GPU to access external memory.
pvbus_s	slave	PVBus	The interface for the CPU to access the GPU registers.

## Parameters for Mali\_G1

### **altcmdline**

Alternate command line for the GPU model. If used, mode is ignored.

Type: `string`

Default value: ""

### **altmodel**

Path to an alternative GPU model library.

Type: `string`

Default value: ""

### **mode**

GPU Mode. Inputs supported: [ca, fast, turbo, turbo\_only].

Type: `string`

Default value: "fast"

### **revision**

Hardware revision. Changing this parameter aligns the behavior of the model with the hardware of specified revision.

Type: `string`

Default value: "r0p0"

### **turbo\_threads**

Number of threads used.

Type: `uint32_t`

Default value: 0x0

## 3.257 Mali\_G71

Defined in `LISA/Mali_G71.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Limitations

- The model does not support Armv8-style page tables.
- The model does not execute GPU shader programs.
- The model does not validate all register values or job descriptors.
- The model does not implement the CNTVALUEB port, and the GPU timestamp does not align with the A-Profile CNTPCT\_ELO system register.

This model outputs values for all Mali PMU hardware counters. These values are not meaningful. Their purpose is to enable you to use the model early in the development process to check that your system is correctly configured to support performance analysis tools, for instance Streamline.

The counter is a 32-bit bitfield, with the following bit assignments:

#### [31:28]

Identifies which Mali GPU instance generated this value, typically zero for a system with a single GPU.

#### [27:24]

Identifies which hardware block is the source of the counter. It can have one of the following values:

**0**

Job manager.

**1**

Tiler.

**2**

L2Cache/Memory system.

**3+**

Shader core.

#### [23:16]

The counter number within the block.

**[15:0]**

Sawtooth counter that counts from 0 to 0xffff and then resets to 0. This value ensures that consecutive captures show different values and allows you to check that counter values are changing over time.



- These counter values might change in future.
- The model always generates counter values, even if the GPU is idle.
- Streamline does not display the precise values that the model outputs because it needs to correct them for sampling frequency and other profiling effects. However, their size relative to each other is correct. For example, counters from the job manager are always smaller than those from the memory system.

**Iris and MTI instances for Mali\_G71**

This model has the following Iris instances:

Name	Instance type
Mali_G71	Mali_G71
Mali_G71.ExportTest.Mali_G71[0].pvbusmaster	PVBusMaster
Mali_G71.busmaster	PVBusMaster
Mali_G71.busslave	PVBusSlave

This model has the following MTI trace components:

Name	Component type
Mali_G71	Mali_G71
Mali_G71.ExportTest.Mali_G71[0].pvbusmaster	PVBusMaster
Mali_G71.busmaster	PVBusMaster
Mali_G71.busslave	PVBusSlave

**Ports for Mali\_G71**

Port	Direction	Protocol	Description
gpu_reset	slave	Signal	Resets the GPU at the input Set. The available inputs are Set and Clear.
irq_gpu	master	Signal	The interrupt signal generated from the GPU.
irq_job	master	Signal	The interrupt signal generated from the Job Manager on the GPU.
irq_mmu	master	Signal	The interrupt signal generated from the MMU on the GPU.
prot_mode_m0	master	Signal	Indicates the current state of the GPU. When it outputs: Set: The GPU is in protected mode. Clear: The GPU is in normal mode. Typically, connect this signal to the LabellerForGPUProtMode component. This enables the labeller to add a Non-Secure Access ID to the outgoing transaction on the labeller's pvbus_m port.
pvbus_m	master	PVBus	The interface for the GPU to access external memory.
pvbus_s	slave	PVBus	The interface for the CPU to access the GPU registers.

## Parameters for Mali\_G71

### revision

Revision of the RTL that the model represents. Valid values: r0p0.

Type: `string`

Default value: `"r0p0"`

## 3.258 Mali\_G710

Defined in `LISA/Mali_G710.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### About Mali\_G710

The Mali\_G710 and Mali\_G715 components model the Arm Mali G710 and G715 GPUs respectively, which implement the Valhall architecture. They are among the first Mali GPUs to implement a Command Stream Frontend (CSF), which is a combination of hardware and firmware that helps to offload work from the Mali driver that is running on the CPU.

With the Mali G710 and G715 Fast Models, you can simulate:

- User space and kernel space driver components of the Arm Mali Driver Development kit (Mali DDK), when built to target Mali G710 or G715 under Android and Linux graphics stacks.
- Mali G710 or G715 firmware binaries.

These are functional models that do not attempt to execute GPU shader programs. They are sufficient to simulate a Mali driver, but rather than deliver fully rendered images to memory, they write a randomly-generated color gradient pattern to the framebuffer.

The changing frames can be used to prove that the graphics stack is correctly configured to allow the Mali GPU to write to a user-visible framebuffer.

### Configuring the driver for use with the model

To account for timing differences between real hardware and the Fast Model, it is necessary to modify the following runtime settings in the Mali driver:

#### **csf\_firmware\_boot\_timeout\_ms**

Overrides the minimum timeout value for loading firmware into the model.



**Type**

Kernel module parameter

**Recommended value**

10000

**reset\_timeout**

Overrides the minimum timeout when waiting for operations on the GPU to finish.

**Type**

Sysfs parameter

**Recommended value**

1000000

**fw\_timeout**

Overrides the minimum timeout when waiting for operations on the GPU to finish.

**Type**

Sysfs parameter

**Recommended value**

1000000

The following commands are a snippet from an Android RC file that is an example of how a filesystem can be configured to correctly load the Mali driver when running on a Fast Models platform:

```
insmod /vendor/lib/modules/mali_kbase.ko csf_firmware_boot_timeout_ms=100000  
wait /dev/mali0  
chmod 0666 /dev/mali0  
wait /sys/class/misc/mali0/device/reset_timeout  
write /sys/class/misc/mali0/device/reset_timeout 1000000  
write /sys/class/misc/mali0/device/fw_timeout 1000000
```

**Limitations**

- The model does not support debug access to registers through Iris-enabled debuggers.
- The timing data observed from the model does not correspond in any way to real hardware.
- The model does not expose any MTI targets.
- The model is not supported on Windows hosts.
- The model does not implement the CNTVALUEB port, and the GPU timestamp does not align with the A-Profile CNTPCT\_ELO system register.
- Turbo mode is not supported.
- Not all performance counters are updated, and those that are may not always report accurate values.
- The model may not cleanly recover from GPU reset, suspend, or resume operations. To work around this, it is recommended that any GPU governors are left in an always-on mode, and

the operating system is configured to avoid GPU resets due to lack of progress. Driver timeout settings may also need adjustment to accomodate slower processing.

## Iris and MTI instances for Mali\_G710

This model has the following Iris instances:

Name	Instance type
Mali_G710	Mali_G710
Mali_G710.ExportTest.Mali_G710[0].pvbusmaster	PVBusMaster
Mali_G710.busmaster	PVBusMaster
Mali_G710.subordinate	PVBusSlave

This model has the following MTI trace components:

Name	Component type
Mali_G710	Mali_G710
Mali_G710.ExportTest.Mali_G710[0].pvbusmaster	PVBusMaster
Mali_G710.busmaster	PVBusMaster
Mali_G710.subordinate	PVBusSlave

## Ports for Mali\_G710

Port	Direction	Protocol	Description
gpu_reset	slave	Signal	Resets the GPU at the input Set. The available inputs are Set and Clear.
irq_gpu	master	Signal	The interrupt signal generated from the GPU.
irq_job	master	Signal	The interrupt signal generated from the Job Manager on the GPU.
irq_mmu	master	Signal	The interrupt signal generated from the MMU on the GPU.
prot_mode_m0	master	Signal	Indicates the current state of the GPU. When it outputs: Set: The GPU is in protected mode. Clear: The GPU is in normal mode. Typically, connect this signal to the LabellerForGPUProtMode component. This enables the labeller to add a Non-Secure Access ID to the outgoing transaction on the labeller's pvbus_m port.
pvbus_m	master	PVBus	The interface for the GPU to access external memory.
pvbus_s	slave	PVBus	The interface for the CPU to access the GPU registers.

## Parameters for Mali\_G710

### altcmdline

Alternate command line for the GPU model. If used, mode is ignored.

Type: string

Default value: ""

### altmodel

Path to an alternative GPU model library.

Type: `string`

Default value: `""`

#### **mode**

GPU Mode. Inputs supported: `[ca, fast]`.

Type: `string`

Default value: `"fast"`

#### **revision**

Hardware revision. Changing this parameter aligns the behavior of the model with the hardware of specified revision.

Type: `string`

Default value: `"r0p0"`

## 3.259 Mali\_G715

Defined in `LISA/Mali_G715.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0 r0p1 r1p0 r1p1 r1p2 r1p3	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### About Mali\_G715

The Mali\_G710 and Mali\_G715 components model the Arm Mali G710 and G715 GPUs respectively, which implement the Valhall architecture. They are among the first Mali GPUs to implement a Command Stream Frontend (CSF), which is a combination of hardware and firmware that helps to offload work from the Mali driver that is running on the CPU.

With the Mali G710 and G715 Fast Models, you can simulate:

- User space and kernel space driver components of the Arm Mali Driver Development kit (Mali DDK), when built to target Mali G710 or G715 under Android and Linux graphics stacks.
- Mali G710 or G715 firmware binaries.

These are functional models that do not attempt to execute GPU shader programs. They are sufficient to simulate a Mali driver, but rather than deliver fully rendered images to memory, they write a randomly-generated color gradient pattern to the framebuffer.

The changing frames can be used to prove that the graphics stack is correctly configured to allow the Mali GPU to write to a user-visible framebuffer.

## Configuring the driver for use with the model

To account for timing differences between real hardware and the Fast Model, it is necessary to modify the following runtime settings in the Mali driver:

### **csf\_firmware\_boot\_timeout\_ms**

Overrides the minimum timeout value for loading firmware into the model.

#### **Type**

Kernel module parameter

#### **Recommended value**

10000

### **reset\_timeout**

Overrides the minimum timeout when waiting for operations on the GPU to finish.

#### **Type**

Sysfs parameter

#### **Recommended value**

1000000

### **fw\_timeout**

Overrides the minimum timeout when waiting for operations on the GPU to finish.

#### **Type**

Sysfs parameter

#### **Recommended value**

1000000

The following commands are a snippet from an Android RC file that is an example of how a filesystem can be configured to correctly load the Mali driver when running on a Fast Models platform:

```
insmod /vendor/lib/modules/mali_kbase.ko csf_firmware_boot_timeout_ms=100000
wait /dev/mali0
chmod 0666 /dev/mali0
wait /sys/class/misc/mali0/device/reset_timeout
write /sys/class/misc/mali0/device/reset_timeout 1000000
write /sys/class/misc/mali0/device/fw_timeout 1000000
```

## Limitations

- The model does not support debug access to registers through Iris-enabled debuggers.
- The timing data observed from the model does not correspond in any way to real hardware.
- The model does not expose any MTI targets.
- The model is not supported on Windows hosts.

- The model does not implement the CNTVALUEB port, and the GPU timestamp does not align with the A-Profile CNTPCT\_ELO system register.
- Turbo mode is not supported.
- Not all performance counters are updated, and those that are may not always report accurate values.
- The model may not cleanly recover from GPU reset, suspend, or resume operations. To work around this, it is recommended that any GPU governors are left in an always-on mode, and the operating system is configured to avoid GPU resets due to lack of progress. Driver timeout settings may also need adjustment to accomodate slower processing.

## Iris and MTI instances for Mali\_G715

This model has the following Iris instances:

Name	Instance type
Mali_G715	Mali_G715
Mali_G715.ExportTest.Mali_G715[0].pvbusmaster	PVBusMaster
Mali_G715.busmaster	PVBusMaster
Mali_G715.subordinate	PVBusSlave

This model has the following MTI trace components:

Name	Component type
Mali_G715	Mali_G715
Mali_G715.ExportTest.Mali_G715[0].pvbusmaster	PVBusMaster
Mali_G715.busmaster	PVBusMaster
Mali_G715.subordinate	PVBusSlave

## Ports for Mali\_G715

Port	Direction	Protocol	Description
gpu_reset	slave	Signal	Resets the GPU at the input Set. The available inputs are Set and Clear.
irq_gpu	master	Signal	The interrupt signal generated from the GPU.
irq_job	master	Signal	The interrupt signal generated from the Job Manager on the GPU.
irq_mmu	master	Signal	The interrupt signal generated from the MMU on the GPU.
prot_mode_m0	master	Signal	Indicates the current state of the GPU. When it outputs: Set: The GPU is in protected mode. Clear: The GPU is in normal mode. Typically, connect this signal to the LabellerForGPUProtMode component. This enables the labeller to add a Non-Secure Access ID to the outgoing transaction on the labeller's pvbus_m port.
pvbus_m	master	PVBus	The interface for the GPU to access external memory.
pvbus_s	slave	PVBus	The interface for the CPU to access the GPU registers.

## Parameters for Mali\_G715

### altcmdline

Alternate command line for the GPU model. If used, mode is ignored.

Type: `string`

Default value: `""`

**altmodel**

Path to an alternative GPU model library.

Type: `string`

Default value: `""`

**mode**

GPU Mode. Inputs supported: `[ca, fast]`.

Type: `string`

Default value: `"fast"`

**revision**

Hardware revision. Changing this parameter aligns the behavior of the model with the hardware of specified revision.

Type: `string`

Default value: `"r0p0"`

### 3.260 Mali\_G720

Defined in `LISA/Mali_G720.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0 r0p1	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

**About Mali\_G720**

This model is the first of a new generation of fully-functional GPU models that generate correct output without the assistance of Generic Graphics Accelerator (GGA).

It supports `x86_64` and `AArch64` hosts running a supported version of Linux, as listed in Requirements for Fast Models in the [Fast Models User Guide](#). It does not support Windows hosts.

`$(PVLIB_HOME)/LISA/Mali_G720.lisa` requires a clock input running at around 500MHz to keep CPU and GPU performance aligned. This avoids software timeouts during slow-running GPU operations.

From a Fast Models perspective, the differences between Arm Mali-G720, Arm Immortalis-G720, and Mali-G620 are minor. The Fast Model does not model the caches and the core count is transparent to the user. Ray tracing is not currently modeled.

## Operating modes

The model supports multiple operating modes which trade performance against functional accuracy. The operating mode is selected using the `mode` parameter.

### Fast mode

The default and recommended operating mode. It provides the highest level of functional accuracy across graphics and compute workloads and is the most robust option for general use, validation, and software bring-up.

### Turbo mode

Enables a just-in-time execution mechanism which significantly improves performance, particularly for compute-focused workloads. This mode is still under active development and is not yet a fully general-purpose execution mode. Compute workloads are expected to work correctly in turbo mode, and generally see the largest performance benefit. Graphics workloads may function in simple cases, but coverage is incomplete and failures are possible.

Turbo mode has known limitations and is not fully validated across all APIs. For workloads requiring stability or broad graphics support, fast mode is recommended.

## Limitations

- The model does not support debug access to registers through Iris-enabled debuggers.
- The timing data observed from the model does not correspond in any way to real hardware.
- The model is not supported on Windows hosts.
- Support for turbo mode is deprecated and may be removed in a future release.
- Not all performance counters are updated, and those that are may not always report accurate values.
- The model may not cleanly recover from GPU reset, suspend, or resume operations. To work around this, it is recommended that any GPU governors are left in an always-on mode, and the operating system is configured to avoid GPU resets due to lack of progress. Driver timeout settings may also need adjustment to accomodate slower processing.

## Iris and MTI instances for Mali\_G720

This model has the following Iris instances:

Name	Instance type
Mali_G720	Mali_G720
Mali_G720.ExportTest.Mali_G720[0].pvbusmaster	PVBusMaster
Mali_G720.busmaster	PVBusMaster

Name	Instance type
Mali_G720.subordinate	PVBusSlave

This model has the following MTI trace components:

Name	Component type
Mali_G720	Mali_G720
Mali_G720.ExportTest.Mali_G720[0].pvbusmaster	PVBusMaster
Mali_G720.busmaster	PVBusMaster
Mali_G720.subordinate	PVBusSlave

### Ports for Mali\_G720

Port	Direction	Protocol	Description
clk_in	slave	ClockSignal	-
cntvalueb	slave	CounterInterface	Interface to the SoC-level Generic Counter module that is used to source the global system timestamp value. When the GPU component is integrated into an A-Profile compliant system, this port must be connected for the GPU and CPU timestamps to be synchronized. Without the connection, the GPU timestamp remains constant. In this case, it is recommended to configure the GPU to use its own internal clock as the reference.
gpu_reset	slave	Signal	Resets the GPU at the input Set. The available inputs are Set and Clear.
irq_gpu	master	Signal	The interrupt signal generated from the GPU.
irq_job	master	Signal	The interrupt signal generated from the Job Manager on the GPU.
irq_mmu	master	Signal	The interrupt signal generated from the MMU on the GPU.
prot_mode_m0	master	Signal	Indicates the current state of the GPU. When it outputs: Set: The GPU is in protected mode. Clear: The GPU is in normal mode. Typically, connect this signal to the LabellerForGPUProtMode component. This enables the labeller to add a Non-Secure Access ID to the outgoing transaction on the labeller's pvbus_m port.
pvbus_m	master	PVBus	The interface for the GPU to access external memory.
pvbus_s	slave	PVBus	The interface for the CPU to access the GPU registers.

### Parameters for Mali\_G720

#### **altcmdline**

Alternate command line for the GPU model. If used, mode is ignored.

Type: string

Default value: ""

#### **altmodel**

Path to an alternative GPU model library.

Type: string

Default value: ""



**mode**  
GPU Mode. Inputs supported: [ca, fast, turbo, turbo\_only].  
  
Type: `string`  
  
Default value: `"fast"`

**revision**  
Hardware revision. Changing this parameter aligns the behavior of the model with the hardware of specified revision.  
  
Type: `string`  
  
Default value: `"r0p0"`

**turbo\_threads**  
Number of threads used.  
  
Type: `uint32_t`  
  
Default value: `0x0`

### 3.261 Mali\_G725

Defined in `LISA/Mali_G725.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0 r0p1	Preliminary support

For an explanation of the quality levels, see [Quality level definitions](#).

#### About Mali\_G725

This component is a model of the Arm Mali-G725 GPU, fully capable of executing shaders and producing graphical or compute outputs.

It supports x86\_64 and AArch64 hosts running a supported version of Linux, as listed in Requirements for Fast Models in the [Fast Models User Guide](#). It does not support Windows hosts.

`$(PVLIB_HOME)/LISA/Mali_G725.lisa` requires a clock input running at around 500MHz to keep CPU and GPU performance aligned. This avoids software timeouts during slow-running GPU operations.

From a Fast Models perspective, the differences between Mali-G725, Arm Immortalis-G925, and Mali-G625 are minor. The Fast Model does not model the caches and the core count is transparent to the user. Ray tracing is not currently modeled.

## Limitations

- The model does not support debug access to registers through Iris-enabled debuggers.
- The timing data observed from the model does not correspond in any way to real hardware.
- The model is not supported on Windows hosts.
- Turbo mode is not supported.
- Not all performance counters are updated, and those that are may not always report accurate values.
- The model may not cleanly recover from GPU reset, suspend, or resume operations. To work around this, it is recommended that any GPU governors are left in an always-on mode, and the operating system is configured to avoid GPU resets due to lack of progress. Driver timeout settings may also need adjustment to accomodate slower processing.

## Iris and MTI instances for Mali\_G725

This model has the following Iris instances:

Name	Instance type
Mali_G725	Mali_G725
Mali_G725.ExportTest.Mali_G725[0].pvbusmaster	PVBusMaster
Mali_G725.busmaster	PVBusMaster
Mali_G725.subordinate	PVBusSlave

This model has the following MTI trace components:

Name	Component type
Mali_G725	Mali_G725
Mali_G725.ExportTest.Mali_G725[0].pvbusmaster	PVBusMaster
Mali_G725.busmaster	PVBusMaster
Mali_G725.subordinate	PVBusSlave

## Ports for Mali\_G725

Port	Direction	Protocol	Description
clk_in	slave	ClockSignal	-
cntvalueb	slave	CounterInterface	Interface to the SoC-level Generic Counter module that is used to source the global system timestamp value. When the GPU component is integrated into an A-Profile compliant system, this port must be connected for the GPU and CPU timestamps to be synchronized. Without the connection, the GPU timestamp remains constant. In this case, it is recommended to configure the GPU to use its own internal clock as the reference.
gpu_reset	slave	Signal	Resets the GPU at the input Set. The available inputs are Set and Clear.
irq_gpu	master	Signal	The interrupt signal generated from the GPU.
irq_job	master	Signal	The interrupt signal generated from the Job Manager on the GPU.

Port	Direction	Protocol	Description
irq_mmu	master	Signal	The interrupt signal generated from the MMU on the GPU.
prot_mode_m0	master	Signal	Indicates the current state of the GPU. When it outputs: Set: The GPU is in protected mode. Clear: The GPU is in normal mode. Typically, connect this signal to the LabellerForGPUProtMode component. This enables the labeller to add a Non-Secure Access ID to the outgoing transaction on the labeller's pvbush_m port.
pvbus_m	master	PVBus	The interface for the GPU to access external memory.
pvbus_s	slave	PVBus	The interface for the CPU to access the GPU registers.

## Parameters for Mali\_G725

### **altcmdline**

Alternate command line for the GPU model. If used, mode is ignored.

Type: `string`

Default value: `""`

### **altmodel**

Path to an alternative GPU model library.

Type: `string`

Default value: `""`

### **mode**

GPU Mode. Inputs supported: [ca, fast].

Type: `string`

Default value: `"fast"`

### **revision**

Hardware revision. Changing this parameter aligns the behavior of the model with the hardware of specified revision.

Type: `string`

Default value: `"r0p0"`

## 3.262 Mali\_G76

Defined in `LISA/Mali_G76.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

## Limitations

- The model does not support Armv8-style page tables.
- The model does not execute GPU shader programs.
- The model does not validate all register values or job descriptors.
- The model does not implement the CNTVALUEB port, and the GPU timestamp does not align with the A-Profile CNTPCT\_ELO system register.

This model outputs values for all Mali PMU hardware counters. These values are not meaningful. Their purpose is to enable you to use the model early in the development process to check that your system is correctly configured to support performance analysis tools, for instance Streamline.

The counter is a 32-bit bitfield, with the following bit assignments:

### [31:28]

Identifies which Mali GPU instance generated this value, typically zero for a system with a single GPU.

### [27:24]

Identifies which hardware block is the source of the counter. It can have one of the following values:

**0**

Job manager.

**1**

Tiler.

**2**

L2Cache/Memory system.

**3+**

Shader core.

### [23:16]

The counter number within the block.

### [15:0]

Sawtooth counter that counts from 0 to 0xffff and then resets to 0. This value ensures that consecutive captures show different values and allows you to check that counter values are changing over time.



- These counter values might change in future versions.
- The model always generates counter values, even if the GPU is idle.
- Streamline does not display the precise values that the model outputs because it needs to correct them for sampling frequency and other profiling effects. However, their size relative to each other is correct. For example, counters from the job manager are always smaller than those from the memory system.

## Iris and MTI instances for Mali\_G76

This model has the following Iris instances:

Name	Instance type
Mali_G76	Mali_G76
Mali_G76.ExportTest.Mali_G76[0].pvbusmaster	PVBusMaster
Mali_G76.busmaster	PVBusMaster
Mali_G76.busslave	PVBusSlave

This model has the following MTI trace components:

Name	Component type
Mali_G76	Mali_G76
Mali_G76.ExportTest.Mali_G76[0].pvbusmaster	PVBusMaster
Mali_G76.busmaster	PVBusMaster
Mali_G76.busslave	PVBusSlave

## Ports for Mali\_G76

Port	Direction	Protocol	Description
gpu_reset	slave	Signal	Resets the GPU at the input Set. The available inputs are Set and Clear.
irq_gpu	master	Signal	The interrupt signal generated from the GPU.
irq_job	master	Signal	The interrupt signal generated from the Job Manager on the GPU.
irq_mmu	master	Signal	The interrupt signal generated from the MMU on the GPU.
prot_mode_m0	master	Signal	Indicates the current state of the GPU. When it outputs: Set: The GPU is in protected mode. Clear: The GPU is in normal mode. Typically, connect this signal to the LabellerForGPUProtMode component. This enables the labeller to add a Non-Secure Access ID to the outgoing transaction on the labeller's pvbus_m port.
pvbus_m	master	PVBus	The interface for the GPU to access external memory.
pvbus_s	slave	PVBus	The interface for the CPU to access the GPU registers.

## Parameters for Mali\_G76

### revision

Revision of the RTL that the model represents. Valid values: r0p0.

Type: string

Default value: "r0p0"

## 3.263 Mali\_G78AE

Defined in `LISA/Mali_G78AE.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0 r0p1 r0p2	Preliminary support

For an explanation of the quality levels, see [Quality level definitions](#).

### About Mali\_G78AE

The Mali\_G78AE component models the Arm Mali G78AE, which implements the Valhall architecture, and is the first Mali GPU designed specifically to target automotive use cases.

The model implements the Partition Manager, enabling up to 4 independent partitions running workloads at the same time, while being accessed by up to 16 virtual machines.

The Mali\_G78AE model is functional, capable of executing GPU shader programs and producing graphical or compute outputs. It requires a clock input running at around 500MHz to keep the CPU and GPU performance aligned, and to avoid software timeouts during slow-running GPU operations.

With this model, and the implemented functionality of the Partition Manager, you can:

- Simulate the entire graphics software stack, including the user space and kernel space driver components of the Arm Mali DDK, and an application that uses a graphics API.
- Verify the integration of the Mali G78AE GPU into the rest of the platform. A complex use case example of this is a system running multiple kernels in Virtual Machines under a Hypervisor, all submitting workloads to the GPU at the same time.

To configure the reference Mali driver for use with the model, we recommend you make some adjustments to the timing parameters of the Mali driver. This is because of timing differences between the real hardware and the Fast Model. Which parameters work best depend on your system, but if you are using the reference Arm implementation of the arbiter from the DDK, one possibility is:

```
insmod mali_kbase.ko gpu_req_timeout=1000
insmod mali_arbiter.ko request_timeout=200 yield_timeout=30
```

### Test applications

The tests have been carried out using the r40p0 release of the Mali DDK. The following applications have been tested and confirmed to work:

- A selection of the Mali DDK integration tests in `product/build-<wsi>/install/bin/`:

- mali\_gles\_integration\_suite
- mali\_cl\_simple\_example
- A selection of lightweight Vulkan examples hosted on [GitHub](#):
  - gears
  - computeparticles, at small particle count
  - texture3d
- A selection of ComputeLibrary examples hosted on [GitHub](#):
  - graph\_lenet
  - graph\_mobilenet\_v2

All of these applications have been successfully tested in a non-virtualised system, on Debian Buster running Linux 4.19.

A subset of these examples have been tested in the following virtualised systems:

- Xen Hypervisor 4.14.1-pre
- Privileged Debian Buster running Linux kernel 4.19, controlling the configuration of partitions, and running the Mali DDK reference arbiter implementation
- Two unprivileged virtual machines, both running Debian Buster, Linux 4.19, both running various applications listed in this section at the same time

## Limitations

- The model does not support debug access to registers through Iris-enabled debuggers.
- It is not supported on Windows hosts.
- It is a functional model and does not simulate performance differences for partitions of different sizes.
- It does not implement the protection \*CHK signals.
- It does not implement Parity/DCLS/CRC faults and fault fingerprints.
- It does not implement different error response modes configurable through the SYSTEM page.
- The model does not implement the CNTVALUEB port, and the GPU timestamp does not align with the A-Profile CNTPCT\_ELO system register.

## Iris and MTI instances for Mali\_G78AE

This model has the following Iris instances:

Name	Instance type
Mali_G78AE	<a href="#">Mali_G78AE</a>
Mali_G78AE.AccessControl	<a href="#">PVBusMapper</a>

This model has the following MTI trace components:

Name	Component type
Mali_G78AE	Mali_G78AE
Mali_G78AE.AccessControl	PVBusMapper

### Ports for Mali\_G78AE

Port	Direction	Protocol	Description
clk_in	slave	ClockSignal	-
gpu_reset_recovery	slave	Signal	-
gpu_reset	slave	Signal	Reset signals
irq_deferred_error	master	Signal	-
irq_groups	master	Signal	-
irq_partitions	master	Signal	Partition Manager irqs
irq_uncorrected_error	master	Signal	-
irq_windows	master	Signal	-
pvbuss_m	master	PVBus	Output to board from GPU
pvbuss_s	slave	PVBus	Slave bus ports (AXI-A through C).
sys_assign_enable	slave	Value	System configuration access control for different ports

### Parameters for Mali\_G78AE

#### labeller\_encoding\_spec

Specification of how the StreamID is encoded into transaction attributes.

Type: string

Default value: "ManagerID64[31:0]=StreamID[31:0]"

#### revision

Hardware revision. Changing this parameter aligns the behavior of the model with the hardware of specified revision.

Type: string

Default value: "r0p0"

## 3.264 Mali\_T624

Defined in LISA/Mali\_T624.lisa.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support



For an explanation of the quality levels, see [Quality level definitions](#).

## Limitations

- The model does not implement the CNTVALUEB port, and the GPU timestamp does not align with the A-Profile CNTPCT\_ELO system register.

## Iris and MTI instances for Mali\_T624

This model has the following Iris instances:

Name	Instance type
Mali_T624	<a href="#">Mali_T624</a>
Mali_T624.ExportTest.Mali_T624[0].pvbusmaster	<a href="#">PVBusMaster</a>
Mali_T624.busmaster	<a href="#">PVBusMaster</a>
Mali_T624.busslave	<a href="#">PVBusSlave</a>

This model has the following MTI trace components:

Name	Component type
Mali_T624	<a href="#">Mali_T624</a>
Mali_T624.ExportTest.Mali_T624[0].pvbusmaster	<a href="#">PVBusMaster</a>
Mali_T624.busmaster	<a href="#">PVBusMaster</a>
Mali_T624.busslave	<a href="#">PVBusSlave</a>

## Ports for Mali\_T624

Port	Direction	Protocol	Description
gpu_reset	slave	<a href="#">Signal</a>	Resets the GPU at the input Set. The available inputs are Set and Clear.
irq_gpu	master	<a href="#">Signal</a>	The interrupt signal generated from the GPU.
irq_job	master	<a href="#">Signal</a>	The interrupt signal generated from the Job Manager on the GPU.
irq_mmu	master	<a href="#">Signal</a>	The interrupt signal generated from the MMU on the GPU.
prot_mode_m0	master	<a href="#">Signal</a>	Indicates the current state of the GPU. When it outputs: Set: The GPU is in protected mode. Clear: The GPU is in normal mode. Typically, connect this signal to the LabellerForGPUProtMode component. This enables the labeller to add a Non-Secure Access ID to the outgoing transaction on the labeller's pvbus_m port.
pvbus_m	master	<a href="#">PVBus</a>	The interface for the GPU to access external memory.
pvbus_s	slave	<a href="#">PVBus</a>	The interface for the CPU to access the GPU registers.

## Parameters for Mali\_T624

### revision

Revision of the RTL that the model represents. Valid values: r0p0.

Type: string

Default value: "r0p0"

## 3.265 MasterClock

Defined in `LISA/MasterClock.lisa`.

### About MasterClock

This component provides a single `clockSignal` output that can be used to drive the `clockSignal` input of `clockDividers`, `clockTimers` and other clocking components.

The rate of the `MasterClock` is not defined because all clocking is relative, but can be considered to be 1 Hz.



**Note**

If the CPU clock frequency is not set to a realistic value, unpredictable behavior might occur, for example the simulation might freeze.

A system might contain more than one `MasterClock`, all of which generate the same `clockSignal` rate.

### Ports for MasterClock

Port	Direction	Protocol	Description
<code>clk_out</code>	master	<code>ClockSignal</code>	Master clock rate.

### Parameters for MasterClock

This component does not have any parameters.

## 3.266 MemoryMappedCounterModule

Defined in `LISA/MemoryMappedCounterModule.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### About MemoryMappedCounterModule

This component must be used by multicore models. It also must be used to run a single core system with a timer that runs at a rate that is different to the input clock to the core.



The component has two bus slave ports because the architecture specification permits you to map each set of registers at different, non-contiguous base addresses.

## Iris and MTI instances for MemoryMappedCounterModule

This model has the following Iris instances:

Name	Instance type
MemoryMappedCounterModule	MemoryMappedCounterModule
MemoryMappedCounterModule.pvbus_control_s[0]	PVBusSlave
MemoryMappedCounterModule.pvbus_read_s[0]	PVBusSlave

This model has the following MTI trace components:

Name	Component type
MemoryMappedCounterModule.pvbus_control_s[0]	PVBusSlave
MemoryMappedCounterModule.pvbus_read_s[0]	PVBusSlave

## Ports for MemoryMappedCounterModule

Port	Direction	Protocol	Description
clk_in	slave	ClockSignal	This clock input determines the frequency of the Physical Count provided to the clusters connected to the cntvalueb port.
cntvalueb	master	CounterInterface	This master port implements a private protocol between the cluster and the MemoryMappedCounterModule. This must be connected to the cntvalueb port on each cluster in the system and to the MemoryMappedCounterModule component.
counter_reset	slave	Signal	Resets when set.
pvbus_control_s	slave	PVBus	This slave port provides memory-mapped read write access to the control registers of the module.
pvbus_read_s	slave	PVBus	This slave port provides memory-mapped read access to the status frame registers.

## Parameters for MemoryMappedCounterModule

### base\_frequency

Reset value for CNTFID0, base frequency in Hz.

Type: uint32\_t

Default value: 100000000

### cntcldr0123\_C

Values to be returned for control-frame CIDR registers.

Type: uint32\_t

Default value: 0

**cntcidr0123\_R**

Values to be returned for read-frame CIDR registers.

Type: uint32\_t

Default value: 0

**cntpidr0123\_C**

Values to be returned for control-frame PIDR registers 0-3.

Type: uint32\_t

Default value: 0

**cntpidr0123\_R**

Values to be returned for read-frame PIDR registers 0-3.

Type: uint32\_t

Default value: 0

**cntpidr4567\_C**

Values to be returned for control-frame PIDR registers 4-7.

Type: uint32\_t

Default value: 0

**cntpidr4567\_R**

Values to be returned for read-frame PIDR registers 4-7.

Type: uint32\_t

Default value: 0

**diagnostics**

Diagnostics.

Type: uint32\_t

Default value: 0

**has\_additional\_registers**

Implements additional REFCLK CNT control registers.

Type: bool

Default value: `false`

**has\_counter\_scaling**

Implements ARMv8.4 generic counter scaling.

Type: `bool`

Default value: `false`

**non\_arch\_start\_at\_default**

Firmware is expected to enable the timer at boot time. However, turning this parameter on is a model-specific way of enabling the counter module out of reset.

Type: `bool`

Default value: `false`

**readonly\_is\_WI**

Ignore (rather than failing) on writes to read-frame.

Type: `bool`

Default value: `false`

### 3.267 MemoryMappedGenericTimer

Defined in `LISA/MemoryMappedGenericTimer.lisa`.

**About MemoryMappedGenericTimer**

ARM Generic Timer.

**Iris and MTI instances for MemoryMappedGenericTimer**

This model has the following Iris instances:

Name	Instance type
<code>MemoryMappedGenericTimer</code>	<code>MemoryMappedGenericTimer</code>
<code>MemoryMappedGenericTimer.busbaseX</code> (where $X = 0-15$ )	<code>PVBusSlave</code>
<code>MemoryMappedGenericTimer.busctlbase</code>	<code>PVBusSlave</code>

This model has the following MTI trace components:

Name	Component type
<code>MemoryMappedGenericTimer.busbaseX</code> (where $X = 0-15$ )	<code>PVBusSlave</code>
<code>MemoryMappedGenericTimer.busctlbase</code>	<code>PVBusSlave</code>

## Ports for MemoryMappedGenericTimer

Port	Direction	Protocol	Description
cntpsirq	master	Signal	-
cntvalueb	slave	CounterInterface	-
pvbus_base_s	slave	PVBus	-
pvbus_ctlbase_s	slave	PVBus	-
pvbus_el0base_s	slave	PVBus	-
timer_reset	slave	Signal	-

## Parameters for MemoryMappedGenericTimer

### bypass\_ctlbase

Bypass CNTBase Access Control. Enable if only timer frame feature is required without CNTBase access control.

Type: `bool`

Default value: `false`

### cntel0acr\_implemented

A bit-field of 8 bits, where bit {n} enables CNTELOACR for timer frame {n}.

Type: `uint8_t`

Default value: 0

### diagnostics

Diagnostics.

Type: `uint32_t`

Default value: 0

### frame\_security

Hard-wired/configurable security for frames (N/S/X, one character per timer frame).

Type: `string`

Default value: `""`

### num\_timers

Number of timer frames.

Type: `uint32_t`

Default value: 1

## 3.268 MemoryMappedGenericWatchdog

Defined in `LISA/MemoryMappedGenericWatchdog.lisa`.

### About MemoryMappedGenericWatchdog

This is a high-level watchdog that generates two interrupts rather than an interrupt followed by a reset.

### Iris and MTI instances for MemoryMappedGenericWatchdog

This model has the following Iris instances:

Name	Instance type
MemoryMappedGenericWatchdog	MemoryMappedGenericWatchdog
MemoryMappedGenericWatchdog.busctlbase	PVBusSlave
MemoryMappedGenericWatchdog.busrefbase	PVBusSlave

This model has the following MTI trace components:

Name	Component type
MemoryMappedGenericWatchdog.busctlbase	PVBusSlave
MemoryMappedGenericWatchdog.busrefbase	PVBusSlave

### Ports for MemoryMappedGenericWatchdog

Port	Direction	Protocol	Description
cntvalueb	slave	CounterInterface	-
ctl_pvbus_s	slave	PVBus	-
ref_pvbus_s	slave	PVBus	-
reset_in	slave	Signal	-
WS0	master	Signal	-
WS1	master	Signal	-

### Parameters for MemoryMappedGenericWatchdog

#### NONSECURE

Non-Secure.

Type: `bool`

Default value: `false`

#### arch\_version

Architecture version. Available 0 and 1.

Type: `uint8_t`

Default value: 0

**diagnostics**

Diagnostics.

Type: `uint32_t`

Default value: 0

**product\_id**

Product Identifier.

Type: `uint8_t`

Default value: 0x0

3.269 MessageHandlingUnit

Defined in `LISA/MHU.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
v2.0	Full support
v2.1	Full support
v3	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

Changes in 11.31.15

The following parameters were added:

- `a_to_b_v3.receiver_legacy_tz_en`
- `a_to_b_v3.sender_legacy_tz_en`
- `a_to_b_v3.support_receiver_rme`
- `a_to_b_v3.support_receiver_tze`
- `a_to_b_v3.support_sender_rme`
- `a_to_b_v3.support_sender_tze`

About MessageHandlingUnit

Message Handling Unit.



## Iris and MTI instances for MessageHandlingUnit

This model has the following Iris instances:

Name	Instance type
MessageHandlingUnit	MessageHandlingUnit
MessageHandlingUnit.a_to_b_v2	MessageHandlingUnitV2
MessageHandlingUnit.a_to_b_v3	MessageHandlingUnitV3
MessageHandlingUnit.version_mapper_a_to_b_rec	PVBusMapper
MessageHandlingUnit.version_mapper_a_to_b_snd	PVBusMapper

This model has the following MTI trace components:

Name	Component type
MessageHandlingUnit.a_to_b_v2	MessageHandlingUnitV2
MessageHandlingUnit.a_to_b_v3	MessageHandlingUnitV3
MessageHandlingUnit.version_mapper_a_to_b_rec	PVBusMapper
MessageHandlingUnit.version_mapper_a_to_b_snd	PVBusMapper

## Ports for MessageHandlingUnit

Port	Direction	Protocol	Description
pvbus_s_rec	slave	PVBus	-
pvbus_s_snd	slave	PVBus	-
rec_combined_channel_irq_out	master	Signal	-
rec_combined_irq_out	master	Signal	-
rec_reset_in	slave	Signal	-
snd_combined_channel_irq_out	master	Signal	-
snd_combined_irq_out	master	Signal	-
snd_reset_in	slave	Signal	-

## Parameters for MessageHandlingUnit

### NUM\_DB\_CH

Number of doorbell channels.

Type: uint32\_t

Default value: 1

### NUM\_FAST\_CH

Number of fast channels.

Type: uint32\_t

Default value: 1

**a\_to\_b\_v2.NUM\_CH**

Number of device channels.

Type: uint32\_t

Default value: 1

**a\_to\_b\_v2.minor\_revision**

MHUV2 minor revision.

Type: uint32\_t

Default value: 0

**a\_to\_b\_v2.product\_id**

MHU part number.

Type: uint32\_t

Default value: 0

**a\_to\_b\_v3.NUM\_DB\_CH**

Number of doorbell channels.

Type: uint32\_t

Default value: 1

**a\_to\_b\_v3.NUM\_FAST\_CH**

Number of Fast Channels.

Type: uint32\_t

Default value: 1

**a\_to\_b\_v3.NUM\_FIFO\_CH**

Number of FIFO Channels.

Type: uint32\_t

Default value: 1

**a\_to\_b\_v3.auto\_op\_full**

AutoOp mode - AutoOp(min) == false, AutoOp(full) == true - default: AutoOp(min).

Type: bool

Default value: false

**a\_to\_b\_v3.diagnostics**

Diagnostics 0:FATAL 1:ERROR 2:WARNING 3:INFO 4:DEBUG, Default:WARNING(2).

Type: `uint8_t`

Default value: 2

**a\_to\_b\_v3.fast\_ch\_group\_int\_enable**

Fast Channel group interrupts enable, default=false.

Type: `bool`

Default value: `false`

**a\_to\_b\_v3.fast\_ch\_n\_per\_group**

Fast Channel num channels per group, default=1.

Type: `uint32_t`

Default value: 1

**a\_to\_b\_v3.fast\_ch\_num\_groups**

Fast Channel num of groups, default=1.

Type: `uint32_t`

Default value: 1

**a\_to\_b\_v3.fast\_ch\_word\_size**

Fast Channel word size 32bit or 64bit, default=32.

Type: `uint32_t`

Default value: 32

**a\_to\_b\_v3.fifo\_depth**

Depth of the FIFO = `fifo_depth + 1`.

Type: `uint16_t`

Default value: 4

**a\_to\_b\_v3.m16ba\_spt**

Mailbox 16 bit access support to FIFO registers.

Type: `"bool"`

Default value: 0

**a\_to\_b\_v3.m32ba\_spt**

Mailbox 32 bit access support to FIFO registers.

Type: "bool"

Default value: 1

**a\_to\_b\_v3.m64ba\_spt**

Mailbox 64 bit access support to FIFO registers.

Type: "bool"

Default value: 0

**a\_to\_b\_v3.m8ba\_spt**

Mailbox 8 bit access support to FIFO registers.

Type: "bool"

Default value: 0

**a\_to\_b\_v3.mhu\_arch\_beta01**

true = Aligns to MHUv3.beta01; false = Aligns to MHUv3.2.

Type: bool

Default value: false

**a\_to\_b\_v3.monolithic**

Monolithic or Distributed MHU - default: monolithic(true).

Type: bool

Default value: true

**a\_to\_b\_v3.p16ba\_spt**

Postbox 16 bit access support to FIFO registers.

Type: "bool"

Default value: 0

**a\_to\_b\_v3.p32ba\_spt**

Postbox 32 bit access support to FIFO registers.

Type: "bool"

Default value: 1

**a\_to\_b\_v3.p64ba\_spt**

Postbox 64 bit access support to FIFO registers.

Type: "bool"

Default value: 0

**a\_to\_b\_v3.p8ba\_spt**

Postbox 8 bit access support to FIFO registers.

Type: "bool"

Default value: 0

**a\_to\_b\_v3.receiver\_legacy\_tz\_en**

Receiver Legacy TrustZone Enable - default: false.

Type: bool

Default value: false

**a\_to\_b\_v3.sender\_legacy\_tz\_en**

Sender Legacy TrustZone Enable - default: false.

Type: bool

Default value: false

**a\_to\_b\_v3.support\_receiver\_rme**

Support RME functionality in receiver, set to true if the underlying platform supports RME.

Type: bool

Default value: false

**a\_to\_b\_v3.support\_receiver\_tze**

Support TrustZone functionality in receiver, set to true if the underlying platform supports TrustZone.

Type: bool

Default value: false

**a\_to\_b\_v3.support\_sender\_rme**

Support RME functionality in sender, set to true if the underlying platform supports RME.

Type: bool

Default value: `false`

### **`a_to_b_v3.support_sender_tze`**

Support TrustZone functionality in sender, set to true if the underlying platform supports TrustZone.

Type: `bool`

Default value: `false`

### **`diagnostics`**

Diagnostics 0==FATAL\_ERROR -> 4==DEBUG.

Type: `uint8_t`

Default value: 2

### **`fast_ch_group_int_enable`**

Fast Channel group interrupts enable, default=false.

Type: `bool`

Default value: `false`

### **`fast_ch_n_per_group`**

Fast Channel num channels per group, default=1.

Type: `uint32_t`

Default value: 1

### **`fast_ch_num_groups`**

Fast Channel num of groups, default=1.

Type: `uint32_t`

Default value: 1

### **`fast_ch_word_size`**

Fast Channel word size 32bit or 64bit, default=32.

Type: `uint32_t`

Default value: 32

### **`major_version`**

MHU major version (default=2).

Type: `uint32_t`

Default value: 2

### **mhu\_arch\_beta01**

true = Aligns to MHUv3.beta01; false = Aligns to MHUv3.2.

Type: `bool`

Default value: false

### **minor\_version**

MHU minor version (default=1).

Type: `uint32_t`

Default value: 1

### **product\_id**

MHU part number.

Type: `uint32_t`

Default value: 0

## 3.270 MessageHandlingUnitV2

Defined in `LISA/MHUv2.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### About MessageHandlingUnitV2

Message Handling Unit Version 2.

### Iris and MTI instances for MessageHandlingUnitV2

This model has the following Iris instances:

Name	Instance type
MessageHandlingUnitV2	<a href="#">MessageHandlingUnitV2</a>

This model has the following MTI trace components:

Name	Component type
MessageHandlingUnitV2	MessageHandlingUnitV2

### Ports for MessageHandlingUnitV2

Port	Direction	Protocol	Description
int_access_nr2r	master	Signal	-
int_access_r2nr	master	Signal	-
mhu_combined_irq	master	Signal	-
mhu_irq	master	Signal	-
mhu_snd_irq	master	Signal	-
pvbuss_rec	slave	PVBus	-
pvbuss_snd	slave	PVBus	-
qchannel_mhu_pwr	slave	PChannel	-
reset_rec	slave	Signal	-
reset_snd	slave	Signal	-
snd_combined_irq	master	Signal	-
wakerequest	master	Signal	-

### Parameters for MessageHandlingUnitV2

#### **NUM\_CH**

Number of device channels.

Type: uint32\_t

Default value: 1

#### **minor\_revision**

MHUv2 minor revision.

Type: uint32\_t

Default value: 0

#### **product\_id**

MHU part number.

Type: uint32\_t

Default value: 0



## 3.271 MessageHandlingUnitV3

Defined in LISA/MHUV3.lisa.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
v3	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following parameters were added:

- receiver\_legacy\_tz\_en
- sender\_legacy\_tz\_en
- support\_receiver\_rme
- support\_receiver\_tze
- support\_sender\_rme
- support\_sender\_tze

### About MessageHandlingUnitV3

Message Handling Unit Version 3.

### Iris and MTI instances for MessageHandlingUnitV3

This model has the following Iris instances:

Name	Instance type
MessageHandlingUnitV3	<a href="#">MessageHandlingUnitV3</a>

This model has the following MTI trace components:

Name	Component type
MessageHandlingUnitV3	<a href="#">MessageHandlingUnitV3</a>

### Ports for MessageHandlingUnitV3

Port	Direction	Protocol	Description
pvbus_s_rec	slave	<a href="#">PVBUS</a>	Register access for Receiver/Mailbox
pvbus_s_snd	slave	<a href="#">PVBUS</a>	Register access for Sender/Postbox
rec_combined_irq_out	master	<a href="#">Signal</a>	All interrupts combined for Receiver/MBX
rec_fast_channel_group_irq_out	master	<a href="#">Signal</a>	Receiver fast channel group interrupts
rec_fast_channel_irq_out	master	<a href="#">Signal</a>	Receiver fast channel interrupts
rec_reset_in	slave	<a href="#">Signal</a>	Reset signal for Receiver/Mailbox

Port	Direction	Protocol	Description
snd_combined_irq_out	master	Signal	All Interrupts combined for Sender/PBX
snd_reset_in	slave	Signal	Reset signal for Sender/Postbox

## Parameters for MessageHandlingUnitV3

### **NUM\_DB\_CH**

Number of doorbell channels.

Type: `uint32_t`

Default value: 1

### **NUM\_FAST\_CH**

Number of Fast Channels.

Type: `uint32_t`

Default value: 1

### **NUM\_FIFO\_CH**

Number of FIFO Channels.

Type: `uint32_t`

Default value: 1

### **auto\_op\_full**

AutoOp mode - AutoOp(min) == false, AutoOp(full) == true - default: AutoOp(min).

Type: `bool`

Default value: `false`

### **diagnostics**

Diagnostics 0:FATAL 1:ERROR 2:WARNING 3:INFO 4:DEBUG, Default:WARNING(2).

Type: `uint8_t`

Default value: 2

### **fast\_ch\_group\_int\_enable**

Fast Channel group interrupts enable, default=false.

Type: `bool`

Default value: `false`

**fast\_ch\_n\_per\_group**

Fast Channel num channels per group, default=1.

Type: uint32\_t

Default value: 1

**fast\_ch\_num\_groups**

Fast Channel num of groups, default=1.

Type: uint32\_t

Default value: 1

**fast\_ch\_word\_size**

Fast Channel word size 32bit or 64bit, default=32.

Type: uint32\_t

Default value: 32

**fifo\_depth**

Depth of the FIFO = fifo\_depth + 1.

Type: uint16\_t

Default value: 4

**m16ba\_spt**

Mailbox 16 bit access support to FIFO registers.

Type: bool

Default value: 0

**m32ba\_spt**

Mailbox 32 bit access support to FIFO registers.

Type: bool

Default value: 1

**m64ba\_spt**

Mailbox 64 bit access support to FIFO registers.

Type: bool

Default value: 0

**m8ba\_spt**

Mailbox 8 bit access support to FIFO registers.

Type: `bool`

Default value: `0`

**mhu\_arch\_beta01**

`true` = Aligns to MHUv3.beta01; `false` = Aligns to MHUv3.2.

Type: `bool`

Default value: `false`

**monolithic**

Monolithic or Distributed MHU - default: `monolithic(true)`.

Type: `bool`

Default value: `true`

**p16ba\_spt**

Postbox 16 bit access support to FIFO registers.

Type: `bool`

Default value: `0`

**p32ba\_spt**

Postbox 32 bit access support to FIFO registers.

Type: `bool`

Default value: `1`

**p64ba\_spt**

Postbox 64 bit access support to FIFO registers.

Type: `bool`

Default value: `0`

**p8ba\_spt**

Postbox 8 bit access support to FIFO registers.

Type: `bool`

Default value: `0`

**receiver\_legacy\_tz\_en**

Receiver Legacy TrustZone Enable - default: false.

Type: `bool`

Default value: `false`

**sender\_legacy\_tz\_en**

Sender Legacy TrustZone Enable - default: false.

Type: `bool`

Default value: `false`

**support\_receiver\_rme**

Support RME functionality in receiver, set to true if the underlying platform supports RME.

Type: `bool`

Default value: `false`

**support\_receiver\_tze**

Support TrustZone functionality in receiver, set to true if the underlying platform supports TrustZone.

Type: `bool`

Default value: `false`

**support\_sender\_rme**

Support RME functionality in sender, set to true if the underlying platform supports RME.

Type: `bool`

Default value: `false`

**support\_sender\_tze**

Support TrustZone functionality in sender, set to true if the underlying platform supports TrustZone.

Type: `bool`

Default value: `false`

## 3.272 NI700

Defined in `LISA/NI700.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Preliminary support
r1p0	Preliminary support
r2p0	Preliminary support

For an explanation of the quality levels, see [Quality level definitions](#).

### About NI700

- Major IP revisions (rX) are modeled and are controlled by the `revision` parameter.
- Minor IP revisions (pY) are not modeled. There are no functional differences in the model between pY revisions, with the exception of the `periph_id2` register.
- To configure the model, you must have installed Arm Socrates. The model's `mesh_config_file` parameter defines the mesh placement of NI700 components. Set it to the name of the yaml configuration file emitted by the Socrates export process. Fast Models requires the configuration file to pass Design Rule Check (DRC) in Socrates and does not support manually editing the configuration file. You must use version r1p5-03rel0 or later of Socrates to generate the file. For more information about Socrates, see [Arm Socrates](#) on Arm Developer. To download it, contact Arm Technical Support.
- Interconnect models do not typically model RTL defects.
- The mapping between the port number for ASNI, AMNI, HSNI, HMNI, and PMNI ports and the xxNI name used in the yml file is its index when all xxNIs of that same type are sorted in ascending alphabetical order. For example, if the yml file has three ASNIs `asni_s100_scp`, `asni_s101_dap`, and `asni_s204_periph0`:
  - `asni_s100_scp` is mapped to `pvbuss_s_asni[0]`
  - `asni_s101_dap` is mapped to `pvbuss_s_asni[1]`
  - `asni_s204_periph0` is mapped to `pvbuss_s_asni[2]`

Similarly:

- AMNIs are mapped to `pvbuss_m_amni`
- HSNIs are mapped to `pvbuss_s_hsni`
- HMNIs are mapped to `pvbuss_m_hmni`
- PMNIs are mapped to `pvbuss_m_pmni`

Additionally, NI700's parser prints the name-to-index mappings when the component parameter `print_parser_log=true`.

**Note**

The PMNI RTL supports up to 16 external interfaces, however the model uses a single PMNI port. Through the single PMNI port, the models support targeting each of the 16 interfaces independently in the SAM.

**Note**

- Access to a non-existent register inside of a CFGNI address space is **WI/RAZ**. This commonly happens when accessing a unused register in an endpoint due to a feature being disabled.
- Access to a non-existent register outside of a CFGNI address space is DECERR. This commonly happens when accessing a location that is outside of the rendered config pages, but inside of the config space declared inside of the yaml. Note that the configuration space can be defined in the yaml larger than what is required for the registers.

The following functionality is expected to work:

- The discovery feature to determine the system address of all nodes.
- Hashed and non-hashed memory regions. They are parsed from the `mesh_config_file`.
- MPAM support. Software must configure MPAM override in ASNI nodes by enabling and configuring the `ASNI_AR_MPAM_OVERRIDE` (0x0E0) and `ASNI_AW_MPAM_OVERRIDE` (0x0E4) registers. The `GT_MPAM_SUPPORT` signal is ignored. Software must configure MPAM support in ASNI nodes by enabling and configuring the Request MPAM Override (0x0E0) register.
- IDM support. The IDM features Access control and Reset control are modeled. Starting in r1p0, non-secure versions of the `ACCESS_STATUS` and `RESET_STATUS` registers are present. The DeviceID and the information whether an xxNI has IDM enabled are parsed from the `mesh_config_file`. When an xxNI is isolated with IDM Access Control or under reset with IDM Reset, all transactions to and from that xxNI are aborted. With respect to IDM reset support, IDM reset signals are modeled and they should be connected to the managed devices that are connected to the respective xxNI port. The register `IDM_RESET_CONTROL` is supported. The target xxNI always enters or exits IDM reset immediately and drives the reset signals accordingly. In register `IDM_RESET_STATUS`, the bitfields `active_write` and `active_read` read always zero. In registers `IDM_RESET_READID` and `IDM_RESET_WRITEID`, the bitfields `vmaster_id` and `master_id` read always zero.
- There are no software functional differences for r2p1 and r2p0 can be used in its place.

**Note**

Unless any feature is listed here as supported, it should be assumed that it is absent from the model.

## Limitations

---



### Note

Issues listed in this section have identifiers of the form [SDDKW-x]. These are for Arm internal references only.

---

- Out of scope:
  - PMU counters are not supported. Counter registers are implemented as **RAZ**.
  - QoS is not supported and all related registers are **RAZ/WI**.
  - Error injection and error generation are not supported. All error registers are **RAZ/WI**.
  - Power, clock, and interrupt signals are not supported.
  - No support for IDM timeout detection.
  - No support for reorder buffers or Cyclic Dependency Avoidance Scheme (CDAS).
- The maximum number of manager Network Interfaces is 127. The maximum tested is 127 AMNIs, 127 PMNIs, and 9 HMNIs.
- The maximum number of subordinate Network Interfaces is 128. The maximum tested is 128 ASNIs and 9 HSNIs.
- The maximum voltage, power, and clock domains of 32 each have not been tested.
- There is no support for 1 stripe target in a group, additional granularities, or the additional stripe group remap functionality described in r2p0 TRM section 2.4.5.
- Remapping features not supported:
  - Tested 5 out of the maximum of 8 remap states.
  - The priority for multiple address remapping states.
  - One target remapped to a different target.
  - A single target remapped to a stripe group.
  - One stripe group remapped to a different stripe group.
  - A stripe group remapped to a single target.
- Stripe features not supported:
  - ASNI striping to an HMNI target has not been tested.
  - Limited testing of stripe groups with different numbers of targets and granularities.
  - Single target stripe.
- The hashed memory regions support is limited by Fast Models DMI. Due to the 4KB memory pages in DMI, granularities smaller than 4KB are not accounted for by the model. Thus, subsequent accesses within a 4KB address range are delivered to the same destination node.
- AMNI nodes in the model are interface-indifferent and registers do not reflect the protocol version.
- r2 CMO Response control is not supported.



- There is no revision string for r2p1. r2p0 is functionally equivalent.
- Hashing of stripe groups is limited to a granularity of 4096B.
- xSNI access to CFGNI is not limited by router connectivity defined in the `mesh_config_file`. It considers only whether the xSNI has the CFGNI target defined in its memory map.
- A reset after model startup does not reset the registers or address remap selections.
- `*_IDM_RESET_STRAP` and its effect on the endpoint soft reset and `IDM_RESET_CONTROL` register is not supported.
- IDM for power domains is not supported.
- No register visibility support for a debugger.
- No software control for the CMO terminate response in register `AMNI_CONFIG_CTL`.

### Iris and MTI instances for NI700

This model has the following Iris instances:

Name	Instance type
NI700	NI700
NI700.decoder	PVBusMapper

This model has the following MTI trace components:

Name	Component type
NI700	NI700
NI700.decoder	PVBusMapper

### Ports for NI700

Port	Direction	Protocol	Description
idm_reset_signal_amni	master	Signal	IDM reset signals to AMNIs.
idm_reset_signal_asni	master	Signal	IDM reset signals to ASNIs.
idm_reset_signal_hmni	master	Signal	IDM reset signals to HMNIs.
idm_reset_signal_hsni	master	Signal	IDM reset signals to HSNIs.
idm_reset_signal_pmni	master	Signal	IDM reset signals to PMNIs.
pdbus_m_amni	master	PVBus	AMNI downstream ports.
pdbus_m_hmni	master	PVBus	HMNI downstream ports.
pdbus_m_pmni	master	PVBus	PMNI downstream ports.
pdbus_s_asni	slave	PVBus	ASNI upstream ports.
pdbus_s_hsni	slave	PVBus	HSNI upstream ports.
reset_in	slave	Signal	Reset signal.

### Parameters for NI700

#### `mesh_config_file`

Name of a file containing mesh placement of NI700 components.

Type: `string`

Default value: `""`

### **mpam\_attributes**

User-defined transform to be applied to bus attributes like `ManagerID`, `ExtendedID` or `UserFlags`, for MPAM Attributes encoded into bus attributes.

For example:

```
ExtendedID[62:55]=MPAM_PMG, ExtendedID[54:39]=MPAM_PARTID, ExtendedID[38]=MPAM_NS
```

An empty string disables MPAM support.

Type: `string`

Default value: `""`

### **periphbase**

Value for PERIPHBASE.

Type: `uint64_t`

Default value: `0xffffffffffffffff`

### **print\_config**

Enables printing the config register addresses.

Type: `bool`

Default value: `false`

### **print\_parser\_log**

Enables printing the yaml config parser log messages.

Type: `bool`

Default value: `false`

### **revision**

Component revision. Currently supports `r0p0`, `r1p0`, `r2p0`.

Type: `string`

Default value: `"r2p0"`

**show\_banner**

Show component banner:

- 0
- supress entire banner
- 1
- suppress config file
- 2+
- show full banner.

Type: uint64\_t

Default value: 2

3.273 NI710AE

Defined in LISA/NI710AE.lisa.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p1	Preliminary support

For an explanation of the quality levels, see [Quality level definitions](#).

About NI710AE

- Major IP revisions (rX) are modeled and are controlled by the model `revision` parameter.
- Minor IP revisions (pY) are not modeled. There are no functional differences in the model between pY revisions, with the exception of the `peripheral_id2` register.
- To configure the model, you must have installed Arm Socrates. The model's `mesh_config_file` parameter defines the network interfaces present as well as their configuration options. Set it to the name of the YAML configuration file emitted by the Socrates export process. Fast Models requires the configuration file to pass Design Rule Check (DRC) in Socrates and does not support manual editing of the configuration file. You must use version r1p7-05 of Socrates to generate the file. For more information about Socrates, see [Arm Socrates](#) on Arm Developer. To download it, contact Arm Technical Support.
- Interconnect models do not typically model RTL defects.
- The mapping between the port number for ASNI, AMNI, HSNI, HMNI, and PMNI ports and the xxNI name used in the yml file is its index when all xxNIs of that same type are sorted in ascending alphabetical order. For example, if the yml file has three ASNIs `asni_s100_scp`, `asni_s101_dap`, and `asni_s204_periph0`:
  - `asni_s100_scp` is mapped to `pvbuss_asni[0]`
  - `asni_s101_dap` is mapped to `pvbuss_asni[1]`

- `asni_s204_periph0` is mapped to `pvbuss_s_asni[2]`

Similarly:

- AMNIs are mapped to `pvbuss_m_amni`
- HSNIs are mapped to `pvbuss_s_hsni`
- HMNIs are mapped to `pvbuss_m_hmni`
- PMNIs are mapped to `pvbuss_m_pmni`

Additionally, the name-to-index mappings are printed when the component parameter `print_parser_log=true`.



Note

The PMNI RTL supports up to 16 external interfaces, however the model uses a single PMNI port. Through the single PMNI port, the models support targeting each of the 16 interfaces independently in the SAM.



Note

- Access to a non-existent register inside of a CFGNI address space is **WI/RAZ**. This commonly happens when accessing a unused register in an endpoint due to a feature being disabled.
- Access to a non-existent register outside of a CFGNI address space is DECERR. This commonly happens when accessing a location that is outside of the rendered config pages, but inside of the config space declared inside of the yaml. Note that the configuration space can be defined in the yaml larger than what is required for the registers.

The following functionality is supported in this release:

- Programmer's view related support:
  - Model supports discovery to determine the system address of the following registers:
    - Global, Voltage, Power, and Clock Domain registers.
    - ASNI, HSNI, AMNI, HMNI, PMNI, and PMU registers.
    - APU subfeature registers.
- Address map-related support:
  - Model supports Static Maps. Hashed and non-hashed memory regions parsed from the `mesh_config_file`.
  - Model supports remap states.
- MPAM support:
  - Software must configure MPAM override in ASNI nodes by enabling and configuring the `ASNI_AR_MPAM_OVERRIDE` (0x0E0) and `ASNI_AW_MPAM_OVERRIDE` (0x0E4) registers.
  - The `GT_MPAM_SUPPORT` signal is ignored.

- Software must configure MPAM support in ASNI nodes by enabling and configuring the Request MPAM Override (0x0E0) register.
- IDM support:
  - The IDM features Access control and Reset control are modeled.
  - The DeviceID and the information whether an xxNI has IDM enabled are parsed from the `mesh_config_file`.
  - When an xxNI is isolated with IDM Access Control or under reset with IDM Reset, all transactions to and from that xxNI are aborted
  - With respect to IDM reset support, IDM reset signals are modeled and they should be connected to the managed devices that are connected to the respective xxNI port.
  - The register `IDM_RESET_CONTROL` is supported.
  - The target xxNI always enters or exits IDM reset immediately and drives the reset signals accordingly.
  - In register `IDM_RESET_STATUS`, the bitfields `active_write` and `active_read` read always zero.
  - In registers `IDM_RESET_READID` and `IDM_RESET_WRITEID`, the bitfields `vmaster_id` and `master_id` read always zero.
- APU support.
- FMU related support:
  - A dedicated APB port is present to access all FMU registers.
  - All FMU registers described in the TRM are implemented and accessible via the APB interface.
  - FMU register protection (TRM section 2.9.8) is supported:
    - Registers are locked after reset and unlocked via a secure write of 0xBE to `FMU_KEY`.
    - 64-bit registers can be written using two consecutive secure 32-bit writes.
  - Error injection via `FMU_SMINJERR` (TRM section 2.9.6) is supported.
  - Error record logging is supported:
    - `FMU_ERR_STATUS`, `FMU_ERR_MISCO`, and `FMU_ERRGSR` are updated when errors are injected.
    - Writing to `FMU_ERR_STATUS.V` clears the V bit and all other writable fields in the register. (Some bits are architecturally **RAZ** and default to 0.)
  - Safety Mechanism configuration is supported.
    - To enable or disable Safety Mechanism reporting (TRM section 2.9.3), follow this sequence:
      1. Select the target endpoint by writing to `FMU_SMINFO`.
      2. Enable or disable Safety Mechanism reporting by writing a bitmask to `FMU_SMEN`.
  - Critical Error Interrupt (CRI) is supported.
    - The `CRI` interrupt can now be triggered through error injection and by configuring the critical error vector.

- Error Recovery Interrupt (ERI) is triggered for uncorrected errors.
- Fault Handling Interrupt (FHI) is triggered for corrected or uncorrected errors.



Unless any feature is listed here as supported, it should be assumed that it is absent from the model.

## Limitations



Issues listed in this section have identifiers of the form [SDDKW-x]. These are for Arm internal references only.

- The following features are out of scope for the Fast Model, and will not be supported:
  - PMU counters are not supported. Counter registers are implemented as **RAZ**.
  - QoS is not supported and all related registers are **RAZ/WI**.
  - Power, clock and interrupt signals are not supported.
  - No support for IDM timeout detection.
  - No support for reorder buffers or Cyclic Dependency Avoidance Scheme (CDAS).
- Programmer's view limitations:
  - [SDDKW-84456] Secure Access Override functionality (`xxnI.secure_access` register) is not supported.
  - A reset after model startup does not reset the registers or address remap selections.
  - AMNI nodes in the model are interface-indifferent and registers do not reflect the protocol version.
  - No register visibility support for a debugger.
  - No software control for the CMO terminate response in register `AMNI_CONFIG_CTL`.
- Configuration and interface limitations:
  - The maximum number of manager Network Interfaces is 127. The maximum tested is 127 AMNIs, 127 PMNIs, and 9 HMNIs.
  - The maximum number of subordinate Network Interfaces is 128. The maximum tested is 128 ASNIs and 9 HSNIs.
  - The maximum voltage, power, and clock domains of 32 each have not been tested.
  - CMO Response control is not supported.
- Address map limitations:
  - There is no support for 1 stripe target in a group, additional granularities, or the additional stripe group remap functionality.

- Remapping features not supported:
  - Tested 5 out of the maximum of 8 remap states.
  - The priority for multiple address remapping states.
  - One target remapped to a different target.
  - A single target remapped to a stripe group.
  - One stripe group remapped to a different stripe group.
  - A stripe group remapped to a single target.
- Stripe features not supported:
  - ASNI striping to an HMNI target has not been tested.
  - Limited testing of stripe groups with different numbers of targets and granularities.
  - Single target stripe.
- The hashed memory regions and strip group granularity support is limited by Fast Models DMI. Due to the 4KB memory pages in DMI, granularities smaller than 4KB are not accounted for by the model. Thus, subsequent accesses within a 4KB address range are delivered to the same destination node.
- IDM limitations:
  - [SDDKW-88881] IDM Functionality is not tested and may not function correctly. Contact Arm Technical Support for support interest.
  - [SDDKW-88881] IDM Registers are accessible but `IDM_DEVICE_ID` is 0 for all IDMs.
  - `*_IDM_RESET_STRAP` and its effect on the endpoint soft reset and `IDM_RESET_CONTROL` register is not supported.
  - IDM for power domains is not supported.
- APU limitations
  - The base and size of address regions must be aligned to 4KiB even when `apuRegion4k` is false. A warning will be given and the region may not function properly due to PVBUS limitations.
  - `APU_ENABLE_RESET_STRAP` is not supported and `APU_CTRL.apu_enable` bit should be used to enable APU instead.
  - [SDDKW-80598] APU Region Locking is currently not supported.
  - APU Model does not enforce programming order described in TRM 102756\_0001\_03 Section 3.4.4.5 “Order of programming for APU address region registers”. The APU Model creates regions from the current values of the APU registers when `APU_CTRL.apu_enable = 1` is written.
  - [SDDKW-87737] APU Reprogramming in the model does not enforce the requirement that `APU_CTRL.apu_enable = 0` is written first before `APU_CTRL.apu_enable = 1` is written to reprogram new regions.
  - [SDDKW-82664] APUID is not read from the transaction attributes, instead the model parameter `apu_subsystem_id` is used to specify the `subsystemID/APUID` of the requestor sending in a transaction through an xSNI.

- FMU limitations
  - `FMU_ERR_CTLR_0` (TRM section 15.11.2) is implemented as a stub.
    - Interrupt masking and enable bits have no effect.
    - All interrupts are always enabled.
  - `FMU_ERR_FR_0` (TRM section 15.11.1) is implemented as a stub.
    - Software cannot query support for features like overflow, corrected error, or deferred error handling.
  - FMU reset behavior (TRM section 2.9.9) is not implemented. Writes to reset-related controls have no effect.
  - RAS error classification is not supported:
    - Deferred Error (DE) is not implemented.
    - `CE` (Corrected Error bit) and `UET` (Uncorrected Error Type) fields in `FMU_ERR_STATUS` are not implemented and always read as 0.
    - RAS error prioritization logic (TRM section 2.9.2) is not modeled.
  - Hardware error detection is not modeled:
    - FMU does not receive error packets from safety mechanisms or protocol checkers internal to the NI710AE Fast Model.
    - APU error detection is not supported.
  - Safety Mechanism injection behavior differs from RTL:
    - In the Fast Model, injection is allowed for all supported Safety Mechanisms (SMs) on any node type, as long as the SM is enabled in `FMU_SMEN`.
    - In the RTL implementation, error injection is more restricted:
      - For FMU node types, only SM[1] (External AMBA Interface Protection) and SM[9] (Internal ERR\_AUB CRC Protection) are accepted.
      - For all other node types, all SMs are supported.
  - Error record lookup during injection is simplified:
    - The model matches error packets using only the `NodeID` and `NodeType`.
    - Domain IDs (`VDID`, `PDID`, `CDID`) are not used for record matching.
  - The model assumes that software performs the full FMU initialization sequence as described in the TRM:
    - Including reading `FMU_ERRDEVID`, iterating over `FMU_ERR_MISC0`, and populating internal discovery state.
    - No automatic endpoint discovery is performed by the model.
  - Writing 1 to clear fields other than the `v` bit (valid) in the `FMU_ERR_STATUS<N>` register is not supported.

## Iris and MTI instances for NI710AE

This model has the following Iris instances:



Name	Instance type
NI710AE	NI710AE
NI710AE.decoder	PVBusMapper

This model has the following MTI trace components:

Name	Component type
NI710AE	NI710AE
NI710AE.decoder	PVBusMapper

## Ports for NI710AE

Port	Direction	Protocol	Description
idm_reset_signal_amni	master	Signal	IDM reset signals to AMNIs.
idm_reset_signal_asni	master	Signal	IDM reset signals to ASNIs.
idm_reset_signal_hmni	master	Signal	IDM reset signals to HMNIs.
idm_reset_signal_hsni	master	Signal	IDM reset signals to HSNIs.
idm_reset_signal_pmni	master	Signal	IDM reset signals to PMNIs.
irq_out_fmu_cri	master	Signal	FMU Critical Interrupt output signal.
irq_out_fmu_eri	master	Signal	FMU Error Record Interrupt output signal.
irq_out_fmu_fault	master	Signal	FMU Fault Interrupt output signal.
pvbus_m_amni	master	PVBus	AMNI downstream ports.
pvbus_m_hmni	master	PVBus	HMNI downstream ports.
pvbus_m_pmni	master	PVBus	PMNI downstream ports.
pvbus_s_asni	slave	PVBus	ASNI upstream ports.
pvbus_s_fmu_apb	slave	PVBus	Dedicated APB port to access FMU registers.
pvbus_s_hsni	slave	PVBus	HSNI upstream ports.
reset_in	slave	Signal	Reset signal.

## Parameters for NI710AE

### apu\_subsystem\_id

APUID/SubsystemID of the component connected to each <x>SNI. Specify the subsystem id of the component connected to each <x>SNI by using a format like:

```
<x>SNI<m>=<subsystemID0>,<y>SNI<n>=<subsystemID1>
```

The subsystemID is assumed to be 0 for any component connected to an <x>SNI that does not appear in this list.

Type: string

Default value: ""

**mesh\_config\_file**

Name of a file containing mesh placement of NI710AE components.

Type: `string`

Default value: `""`

**mpam\_attributes**

User-defined transform to be applied to bus attributes like `ManagerID`, `ExtendedID` Or `UserFlags`, for MPAM Attributes encoded into bus attributes.

For example:

```
ExtendedID[62:55]=MPAM_PMG, ExtendedID[54:39]=MPAM_PARTID, ExtendedID[38]=MPAM_NS
```

An empty string disables MPAM support.

Type: `string`

Default value: `""`

**periphbase**

Value for PERIPHBASE.

Type: `uint64_t`

Default value: `0xffffffffffffffff`

**print\_config**

Enables printing the config register addresses.

Type: `bool`

Default value: `false`

**print\_parser\_log**

Enables printing the yaml config parser log messages.

Type: `bool`

Default value: `false`

**revision**

Component revision. Currently supports r0p1.

Type: `string`

Default value: "r0p1"

### **show\_banner**

Show component banner:

**0**

suppress entire banner

**1**

suppress config file

**2+**

show full banner.

Type: uint64\_t

Default value: 2

## 3.274 NOC\_S3

Defined in `LISA/NOC_S3.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Preliminary support
r1p3	Preliminary support

For an explanation of the quality levels, see [Quality level definitions](#).

### About NOC\_S3

- Major IP revisions (rX) are modeled and are controlled by the model `revision` parameter.
- Minor IP revisions (pY) are not modeled. There are no functional differences in the model between pY revisions, with the exception of the `peripheral_id2` register.
- To configure the model, you must have installed Arm Socrates. The model's `mesh_config_file` parameter defines the network interfaces present as well as their configuration options. Set it to the name of the YAML configuration file emitted by the Socrates export process. Fast Models requires the configuration file to pass Design Rule Check (DRC) in Socrates and does not support manually editing the configuration file. You must use version r1p7-06 or later of Socrates to generate the file. For more information about Socrates, see [Arm Socrates](#) on Arm Developer. To download it, contact Arm Technical Support.
- Interconnect models do not typically model RTL defects.
- The mapping between the port number for ASNI, AMNI, HSNI, HMNI, and PMNI ports and the xxNI name used in the yml file is its index when all xxNIs of that same type are sorted in ascending alphabetical order. For example, if the yml file has three ASNIs `asni_s100_scp`, `asni_s101_dap`, and `asni_s204_periph0`:

- `asni_s100_scp` is mapped to `pvbus_s_asni[0]`
- `asni_s101_dap` is mapped to `pvbus_s_asni[1]`
- `asni_s204_periph0` is mapped to `pvbus_s_asni[2]`

Similarly:

- AMNIs are mapped to `pvbus_m_amni`
- HSNIs are mapped to `pvbus_s_hsni`
- HMNIs are mapped to `pvbus_m_hmni`
- PMNIs are mapped to `pvbus_m_pmni`

Additionally, the name-to-index mappings are printed when the component parameter `print_parser_log=true`.



The PMNI RTL supports up to 16 external interfaces, however the model uses a single PMNI port. Through the single PMNI port, the models support targeting each of the 16 interfaces independently in the SAM.



- Access to a non-existent register inside of a CFGNI address space is **WI/RAZ**. This commonly happens when accessing a unused register in an endpoint due to a feature being disabled.
- Access to a non-existent register outside of a CFGNI address space is DECERR. This commonly happens when accessing a location that is outside of the rendered config pages, but inside of the config space declared inside of the yaml. Note that the configuration space can be defined in the yaml larger than what is required for the registers.

Address map-related support:

- Model supports Static Maps and Programmable Address Maps (PAM).
  - Static Map hashed and non-hashed memory regions are parsed from the `mesh_config_file`.
- (PAM flow) Support for `default_tgt_id` strap to set default `xsni` targets.
- Supports configurable address mask (`cmp_addr_mask_{l,u}`, `htg_addr_mask_{l,u}`).
- Model supports topology parameter `sam/regionCompLSB` to set minimum SAM address granule with values between 4KB and 64KB.
- Model supports `No_Target` as a target in Address Maps.
- Model supports remap states in static map flow.

Programmer's view related support:

- Model supports 4KB and 64KB for topology parameter `configNodeGranularity`.
- Model supports discovery to determine the system address of the following node registers:

- Global, Voltage, Power, and Clock Domain registers.
- ASNI, HSNI, AMNI, HMNI, PMNI, and PMU registers.
- IDM, APU, and PAM subfeature registers.
- FMU and FCU.



Unless any feature is listed here as supported, it should be assumed that it is absent from the model.

## Limitations



Issues listed in this section have identifiers of the form [SDDKW-x]. These are for Arm internal references only.

- The following features are out of scope for the Fast Model, and will not be supported:
  - PMU counters are not supported. Counter registers are implemented as **RAZ**.
  - QoS is not supported and all related registers are **RAZ/WI**.
  - Power, clock and interrupt signals are not supported.
  - No support for AHB Locked transfers.
  - No support for IDM timeout detection.
  - No support for reorder buffers or Cyclic Dependency Avoidance Scheme (CDAS).
- APU limitations [SDDKW-82596]:
  - The base and size of address regions must be aligned to 4KB even when `apuRegion4k` is false. A warning will be given and the region may not function properly due to PVBUS limitations.
  - `APU_ENABLE_RESET_STRAP` is not supported and `APU_CTRL.apu_enable` bit should be used to enable APU instead.
  - APU Region Locking is currently not supported. [SDDKW-80598]
  - `APUID` is not read from the transaction attributes, instead the model parameter `apu_subsystem_id` is used to specify the `subsystemID/APUID` of the requestor sending in a transaction through an xSNI. [SDDKW-81014]
  - If a transaction comes through an xSNI which does not have an APU and is routed to an xMNI which does have one, the APU in the xMNI treats the `APUID` as 0.
  - Since the `subsystemID/APUID` is not encoded in the transaction, components downstream to this model can't know the APUID of the requestor.
- Address map and bus traffic limitations [SDDKW-82595]:

- Default target id cannot be configured through `sam_status` register. [SDDKW-77767] It can be provided through `default_tgtid_strap_i` model parameter.
- Routers are not modeled. `xsn1` access only considers whether the target is defined in the `xsn1`'s memory map. [SDDKW-79760]
- No support for AMNI Address shuttering. [SDDKW-80419]
- Model supports 1, 2, and 4 targets for power-of-two stripe group hashing. No other striping functions/target combinations are supported.
- Hashing of stripe groups limited to a minimum granularity of 4096B. This is a DMI limitation of PVBUS. See [Bus traffic in Fast Models](#) for information.
- No support for exclusive monitoring (ASNI/HSNI/AMNI/HMNI). [SDDKW-79385]
- `xsn1` access to `CFGNI` is not limited by router connectivity defined in the `mesh_config_file`. It considers only whether the `xsn1` has the `CFGNI` target defined in its memory map.
- Stripe limitations
  - ASNI striping to an HMNI target has not been tested.
  - Limited testing of stripe groups with different numbers of targets and granularities.
  - No support for single target stripe.
  - There is no support for additional granularities, or the additional stripe group remap functionality.
  - Limited 8-way striping testing for static address maps. Not supported for programmable address maps.
  - User-defined stripe function is not supported for programmable address maps.
- Remapping features not supported:
  - Tested 5 out of the maximum of 8 remap states.
  - The priority for multiple address remapping states.
  - One target remapped to a different target.
  - A single target remapped to a stripe group.
  - One stripe group remapped to a different stripe group.
  - A stripe group remapped to a single target.
- Programmer's view limitations [SDDKW-82591]:
  - 64-bit register accesses are not supported. [SDDKW-75304]
  - No support for secure/root override for register accesses. [SDDKW-77158]
  - IDM registers are not tested. [SDDKW-77474]
  - No support for `idm_sreset_strap_i` functionality. [SDDKW-80451]
  - AMNI nodes in the model are interface-indifferent and the registers may not reflect the protocol version.
  - Error injection and error generation (RAS) are not supported. All error registers are **RAZ/WI**. [SDDKW-73411]

- No support for MTE.
- ASNI `read_channel_mpam_override` and `write_channel_mpam_override` and the HSNI `MPAM_CONTROL` register fields `*mpam_override_value` are limited to 11 bits. [SDDKW-102675]
- Topology size limitations [SDDKW-82589]:
  - Maximum number of Voltage Domains is 32. The maximum tested is 1.
  - Maximum number of Power Domains is 32. The maximum tested is 2.
  - Maximum number of Clock Domains is 32. The maximum tested is 16.
  - Maximum number of PMU nodes is 32 (1 per Clock Domain). The maximum tested is 15.
  - Maximum number of Subordinate Network Interfaces (SNIs) is 128 (ASNI + HSNI). The maximum tested is 128 ASNIs and 9 HSNI.
  - Maximum number of Manager Network Interfaces (MNI) is 127 (AMNI + HMNI + PMNI). The maximum tested is 127 AMNIs, 127 PMNIs, and 9 HMNIs.
- FMU and FCU limitations:
  - Registers are readable and writeable. No other functionality is modeled.
- CMO limitations:
  - No software control for the CMO terminate response in register `AMNI_CONFIG_CTL`.
  - CMO Response control is not supported.
- A warm reset using static address maps has not been tested. [SDDKW-84005]
- RME limitations:
  - RME disable per interface is not supported. The disable only works when the topology parameter `rmeSupport` is Disabled on all interfaces. Complete support to disable RME such as hiding root override regs has not been modeled or tested. [SDDKW-85413]
  - RME disable does not prevent REALM/ROOT transactions from flowing through interconnect. [SDDKW-81472]
- AMNI nodes in the model are interface-indifferent and registers do not reflect the protocol version.
- A reset after model startup does not reset the registers or address remap selections.
- No register visibility support for a debugger.
- r1 features not supported:
  - AXI-Stream (TSNI and TMNI) is not supported
  - AMNI and HMNI exclusive monitors are not supported
  - Ordered Write Observation (OWO) is not supported
  - Resource Planes (RP) are not supported
  - Unique TX id generator is not supported
  - More than 32 Clock Domains are not supported

## Iris and MTI instances for NOC\_S3

This model has the following Iris instances:

Name	Instance type
NOC_S3	NOC_S3
NOC_S3.decoder	PVBusMapper

This model has the following MTI trace components:

Name	Component type
NOC_S3	NOC_S3
NOC_S3.decoder	PVBusMapper

## Ports for NOC\_S3

Port	Direction	Protocol	Description
idm_reset_signal_amni	master	Signal	IDM reset signals to AMNIs.
idm_reset_signal_asni	master	Signal	IDM reset signals to ASNIs.
idm_reset_signal_hmni	master	Signal	IDM reset signals to HMNIs.
idm_reset_signal_hsni	master	Signal	IDM reset signals to HSNIs.
idm_reset_signal_pmni	master	Signal	IDM reset signals to PMNIs.
pvbus_m_amni	master	PVBus	AMNI downstream ports.
pvbus_m_hmni	master	PVBus	HMNI downstream ports.
pvbus_m_pmni	master	PVBus	PMNI downstream ports.
pvbus_s_asni	slave	PVBus	ASNI upstream ports.
pvbus_s_hsni	slave	PVBus	HSNI upstream ports.
reset_in	slave	Signal	Reset signal.

## Parameters for NOC\_S3

### apu\_subsystem\_id

APUID/SubsystemID of the component connected to each xSNI.

Specify the subsystem id of the component connected to each <x>SNI by using a format like:

```
<x>SNI<m>=<subsystemID0>,<y>SNI<n>=<subsystemID1>
```

The subsystemID is assumed to be 0 for any component connected to an <x>SNI that does not appear in this list.

Type: string

Default value: ""



**default\_tgt\_id\_strap\_i**

Default Target ID input.

Specify the target id for each <x>SNI by using a format like:

```
<x>SNI<m>=<target_id0>,<y>SNI<n>=<target_id1>
```

CFGNI (Configuration Network Interface) is used as the default target if a certain <x>SNI does not appear in the list.

Type: `string`

Default value: `""`

**mesh\_config\_file**

Name of a file containing mesh placement of NOC\_S3 components.

Type: `string`

Default value: `""`

**mpam\_attributes**

User-defined transform to be applied to bus attributes like `ManagerID`, `ExtendedID` Or `UserFlags`, for MPAM Attributes encoded into bus attributes.

For example:

```
ExtendedID[62:55]=MPAM_PMG, ExtendedID[54:39]=MPAM_PARTID, ExtendedID[38]=MPAM_NS
```

An empty string disables MPAM support.

Type: `string`

Default value: `""`

**periphbase**

Value for PERIPHBASE.

Type: `uint64_t`

Default value: `0xffffffffffffffff`

**print\_config**

Enables printing the config register addresses.

Type: `bool`

Default value: `false`

**print\_parser\_log**

Enables printing the yaml config parser log messages.

Type: `bool`

Default value: `false`

**revision**

Component revision. Currently supports r0p0, r1p3.

Type: `string`

Default value: `"r0p0"`

**show\_banner**

Show component banner:

- 0
- supress entire banner
- 1
- suppress config file
- 2+
- show full banner.

Type: `uint64_t`

Default value: `2`

3.275 NonVolatileCounter

Defined in `LISA/NonVolatileCounter.lisa`.

About NonVolatileCounter

Trusted Non-Volatile Counter unit.

Iris and MTI instances for NonVolatileCounter

This model has the following Iris instances:

Name	Instance type
NonVolatileCounter	NonVolatileCounter
NonVolatileCounter.pvbusslave	PVBusSlave

This model has the following MTI trace components:

Name	Component type
NonVolatileCounter.pvbusslave	PVBusSlave

### Ports for NonVolatileCounter

Port	Direction	Protocol	Description
pvbus_s	slave	PVBus	-

### Parameters for NonVolatileCounter

#### **diagnostics**

Diagnostics.

Type: `uint32_t`

Default value: 0

#### **rst\_non\_tz\_fw\_cnt**

Value of NON\_TZ\_FW\_CNT at reset.

Type: `uint32_t`

Default value: 0

#### **rst\_tz\_fw\_cnt**

Value of TZ\_FW\_CNT at reset.

Type: `uint32_t`

Default value: 0

#### **secure**

Instantiate model as Secure (1) or NS (0).

Type: `bool`

Default value: 1

#### **version**

Version of the model functionality. Valid values are r0 and r1.

Type: `string`

Default value: "r0"

## 3.276 OTPW

Defined in `LISA/OTPW.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### About OTPW

One Time Programmable Memory Wrapper.

### Iris and MTI instances for OTPW

This model has the following Iris instances:

Name	Instance type
OTPW	<a href="#">OTPW</a>

### Ports for OTPW

Port	Direction	Protocol	Description
<code>apb_host_in</code>	slave	<a href="#">PVBUS</a>	APB3 Subordinate Interface - Access to OTP memory or OTP emulation memory
<code>apb_otpw_out</code>	master	<a href="#">PVBUS</a>	APB3 Manager Interface - Access to OTP memory
<code>apb_register_in</code>	slave	<a href="#">PVBUS</a>	APB3 Subordinate Interface - Access to OTPW registers
<code>axi_emulated_otpw_out</code>	master	<a href="#">PVBUS</a>	AXI Manager Interface - Access to emulated OTP memory
<code>n_poreset_in</code>	slave	<a href="#">Signal</a>	ICLU power-on reset in
<code>n_warmreset_in</code>	slave	<a href="#">Signal</a>	ICLU power-on warm reset in
<code>otpw_alarm_out</code>	master	<a href="#">Signal</a>	Alarm output destined for the Security Alarm Manager
<code>otpw_int_out</code>	master	<a href="#">Signal</a>	Interrupt output destined for the processor
<code>otpw_otpw_is_ready_out</code>	master	<a href="#">Signal</a>	Status signal indicating that the OTPW is ready

### Parameters for OTPW

#### **GP\_AON\_0\_INIT\_VAL**

Initial value of GP\_AON\_0.

Type: `uint32_t`

Default value: 0

#### **GP\_AON\_1\_INIT\_VAL**

Initial value of GP\_AON\_1.

Type: `uint32_t`

Default value: 0

**OTP\_SIZE\_IN\_WORDS**

The size of the OTP region accessible from the LCM APB-S interface. The maximum size is 60KB.

Type: uint32\_t

Default value: 4096

**diagnostics**

Diagnostics.

Type: uint32\_t

Default value: 2

3.277 OrGate

Defined in LISA/Gate.lisa.

**About OrGate**

This component implements a logical OR of two signal input ports to generate a single output signal. For example, you can use this component to combine two interrupt signals.

**Iris and MTI instances for OrGate**

This model has the following Iris instances:

Name	Instance type
OrGate	OrGate

**Ports for OrGate**

Port	Direction	Protocol	Description
input	slave	Signal	16 input signals to be OR'ed.
output	master	Signal	OR'ed output signal.

**Parameters for OrGate**

This component does not have any parameters.

## 3.278 PASSwitch

Defined in `LISA/PASSwitch.lisa`.

### About PASSwitch

Allow transactions from Realm Management Extension(RME) worlds (realm/root/secure/non\_secure) to be routed separately.

### Iris and MTI instances for PASSwitch

This model has the following Iris instances:

Name	Instance type
PASSwitch	PASSwitch
PASSwitch.mapper	PVBusMapper

This model has the following MTI trace components:

Name	Component type
PASSwitch.mapper	PVBusMapper

### Ports for PASSwitch

Port	Direction	Protocol	Description
control	slave	PASSwitchControl	Controls routing of transactions.
pvbus_m	master	PVBus	Manager ports of PASSwitch.
pvbus_s	slave	PVBus	Subordinate port of PASSwitch.

### Parameters for PASSwitch

#### **non\_secure\_port\_index**

Port index for Non-Secure world transactions to exit or:

- 2  
IGNORE
- 1  
ABORT.

Type: `int32_t`

Default value: 1

#### **non\_secure\_protected\_port\_index**

Port index for Non-Secure Protected world transaction to exit or:

- 2  
IGNORE

**-1**

ABORT.

Type: `int32_t`

Default value: 5

**port\_map\_json**

A JSON value describing ports for different address regions.

The `begin` address and `size` values should be aligned to 4KiB. The format is as follows:

```
[
  {
    "begin": 0x0,
    "size": 0x1000,
    "port": 0
  },
  {
    "begin": 0x20000,
    "size": 0x5000,
    "port": 2
  }
]
```

Type: `string`

Default value: ""

**realm\_port\_index**

Port index for Realm world transactions to exit or:

**-2**

IGNORE

**-1**

ABORT.

Type: `int32_t`

Default value: 3

**root\_port\_index**

Port index for Root world transactions to exit or:

**-2**

IGNORE

**-1**

ABORT.

Type: `int32_t`

Default value: 2

**secure\_port\_index**

Port index for Secure world transactions to exit or:

- 2  
IGNORE
- 1  
ABORT.

Type: int32\_t

Default value: 0

**system\_agent\_port\_index**

Port index for system agent to exit or:

- 2  
IGNORE
- 1  
ABORT.

Type: int32\_t

Default value: 4

3.279 PCIeATC

Defined in LISA/PCIeATC.lisa.

About PCIeATC

This component is for validation only. It is not directly suitable for use as an ATC. It is used for testing the ATC implementation of `pcie_atc_if` produced by `make_PCIeATC_v0()`.

Iris and MTI instances for PCIeATC

This model has the following Iris instances:

Name	Instance type
PCIeATC	<a href="#">validation_atc</a>
PCIeATC.ExportTest.PCIeATC.mapper	<a href="#">PVBUSMapper</a>
PCIeATC.ats[0].pvbusmaster	<a href="#">PVBUSMaster</a>
PCIeATC.pvbus_id_routed_s[0]	<a href="#">PVBUSSlave</a>

This model has the following MTI trace components:



Name	Component type
PCIEATC	atc
PCIEATC.ExportTest.PCIEATC.mapper	PVBusMapper
PCIEATC.ats[0].pvbusmaster	PVBusMaster
PCIEATC.pvbus_id_routed_s[0]	PVBusSlave

## Ports for PCIEATC

Port	Direction	Protocol	Description
atc	slave	PCIEATC_get_if	-
disable_PRI_and_set_RF	master	Signal	This is pulsed (set, then clear) when a condition occurs that causes a Response Failure. The correct response of the PCIe device is to disable PRI and to set the RF bit in the PRI header.
identify	master	SMMUv3AEMIdentifyProtocol	The user has a chance to determine how the substreamid is extracted from the transactions received on pvbus_s by using this port. If it is unimplemented then the ATC will use the default policy identified in SMMUv3_FOR_PCIE.lisa. See How to extract the StreamID and SubstreamID.
pvbus_id_routed_s	slave	PVBus	-
pvbus_m	master	PVBus	-
pvbus_s	slave	PVBus	-
uprgi	master	Signal	This is pulsed (set, then clear) when an Unrecognised PRG Index is received. In a real PCIe device this would set the UPRGI bit in the PRI header.

## Parameters for PCIEATC

### atc\_size

The maximum number of ATC entries. 0 is effectively a large number.

Type: unsigned

Default value: 0

### seed

Seed for a random number generator.

Type: uint64\_t

Default value: 0x12345678

## 3.280 PChannel2SystemC

Defined in `examples/SystemCExport/Bridges/PChannel2SystemC.lisa`.

### About PChannel2SystemC

PChannel to SystemC Converter.

### Iris and MTI instances for PChannel2SystemC

This model has the following Iris instances:

Name	Instance type
PChannel2SystemC	PChannel2SystemC

### Ports for PChannel2SystemC

Port	Direction	Protocol	Description
pchannel	slave	PChannel	-
sc_pchannel	master	SystemCPChannel	-

### Parameters for PChannel2SystemC

This component does not have any parameters.

## 3.281 PL011\_Uart

Defined in `LISA/PL011_Uart.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r1p4	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Using in\_file and in\_file\_escape\_sequence parameters

The UART reads input from `in_file`. If `in_file` contains a line beginning:

```
## WaitForPrompt <something-up-to-end-of-line>
```

then the UART stops reading from `in_file` until the prompt has appeared.

For example, if `in_file` contains the following lines, the UART outputs `1s` only after the root prompt appears:

```
## WaitForPrompt root #
```

ls

**Note**

Use the parameter `in_file_escape_sequence` to set a different escape sequence to `##`.

### Using the `untimed_fifos` parameter

When the `untimed_fifos` parameter is false, characters of serial data are clocked to or from the `SerialData` port at a rate controlled by the `clk_in_ref` clock rate and the `baud-rate-divider` configuration of the UART clock. Enabling `untimed_fifos` permits serial data to be sent or received as fast as it can be generated or consumed. The modem control signals are still generated correctly, so the UART is not able to transmit data faster than the receiving end can handle. For example, `TelnetTerminal` uses the CTS signal to avoid overflowing its TCP/IP buffer. See [TelnetTerminal](#).

### Differences between the model and the RTL

This component does not implement the DMA functionality of the PL011 PrimeCell.

### Iris and MTI instances for `PL011_Uart`

This model has the following Iris instances:

Name	Instance type
<code>PL011_Uart</code>	<code>PL011_Uart</code>
<code>PL011_Uart.busslave</code>	<code>PVBusSlave</code>
<code>PL011_Uart.clk_divider</code>	<code>ClockDivider</code>
<code>PL011_Uart.timer</code>	<code>ClockTimerThread</code>
<code>PL011_Uart.timer.timer</code>	<code>ClockTimerThread64</code>
<code>PL011_Uart.timer.timer.thread</code>	<code>SchedulerThread</code>
<code>PL011_Uart.timer.timer.thread_event</code>	<code>SchedulerThreadEvent</code>

This model has the following MTI trace components:

Name	Component type
<code>PL011_Uart</code>	<code>PL011_Uart</code>
<code>PL011_Uart.busslave</code>	<code>PVBusSlave</code>
<code>PL011_Uart.clk_divider</code>	<code>ClockDivider</code>

### Ports for `PL011_Uart`

Port	Direction	Protocol	Description
<code>clk_in_ref</code>	slave	<code>ClockSignal</code>	Clock input, typically 14.745MHz, which sets the master transmit/receive rate.
<code>intr</code>	master	<code>Signal</code>	Interrupt signal.
<code>pvbus</code>	slave	<code>PVBus</code>	Subordinate port for register access.
<code>serial_out</code>	master	<code>SerialData</code>	Serial input/output and control signals. Used to communicate with a serial device, such as a terminal.

## Parameters for PL011\_Uart

### **baud\_rate**

Baud rate.

Type: `uint32_t`

Default value: 38400

### **clk\_divider.div**

Clock Rate Divider.

Type: `uint64_t`

Default value: 1

### **clk\_divider.mul**

Clock Rate Multiplier.

Type: `uint64_t`

Default value: 1

### **clock\_rate**

Clock rate for PL011.

Type: `uint32_t`

Default value: 14745600

### **diagnostics**

Diagnostics.

Type: `uint8_t`

Default value: 2

### **enable\_dc4**

Enable DC4 commands (try `echo -e "help\024"` in a Linux shell in a serial console).

Type: `bool`

Default value: `true`

### **flow\_ctrl\_mask\_en**

Enable hardware flow control workaround which forcefully disables CTSen and RTSen bits in UARTCR register.

Type: `bool`

Default value: `false`

### **halt**

Halt instead of shutdown for `shutdown_on_eot` and `shutdown_tag`.

Type: `bool`

Default value: `false`

### **in\_file**

Input file for data to be read by the UART.

Type: `string`

Default value: `""`

### **in\_file\_escape\_sequence**

Input file escape sequence.

Type: `string`

Default value: `###`

### **out\_file**

Output file to hold data written by the UART (use '-' to send all output to stdout).

Type: `string`

Default value: `""`

### **revision**

Revision to simulate.

Type: `string`

Default value: `"r1p4"`

### **shutdown\_on\_eot**

Shutdown simulation when a EOT (ASCII 4) char is transmitted (useful for regression tests when semihosting is not available).

Type: `bool`

Default value: `false`

**shutdown\_tag**

Shutdown simulation when a std::string is transmitted.

Type: `string`

Default value: `""`

**toggle\_mti**

Start/stop token for any ToggleMTI source. Argument uses the JSON format: 'START-TOKEN/END-TOKEN' are the corresponding start/stop tokens for toggling the trace plugins. Note that '\n' will be ignored if at start or end of the token. additional information, use 'help' as the value of this parameter.

Type: `string`

Default value: `""`

**uart\_enable**

Enable uart when the system starts up. (clock\_rate and baud\_rate are only valid when this option is enabled.).

Type: `bool`

Default value: `false`

**uart\_fifo\_capacity**

Decides the size of UART fifo.

Type: `uint32_t`

Default value: `32`

**unbuffered\_output**

Unbuffered output.

Type: `bool`

Default value: `false`

**untimed\_fifos**

Ignore the clock rate and transmit/receive serial data immediately.

Type: `bool`

Default value: `true`

## 3.282 PL022\_SSP

Defined in `LISA/PL022_SSP.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r1p4	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Differences between the model and the RTL

Although the PL022\_SSP component has clock input, it is not internally clock-driven. This is different to the hardware.



This component is a preliminary release. It is not a fully-supported peripheral.

### Iris and MTI instances for PL022\_SSP

This model has the following Iris instances:

Name	Instance type
PL022_SSP	PL022_SSP
PL022_SSP.busslave	PVBusSlave
PL022_SSP.prescaler	ClockDivider

This model has the following MTI trace components:

Name	Component type
PL022_SSP.busslave	PVBusSlave
PL022_SSP.prescaler	ClockDivider

### Ports for PL022\_SSP

Port	Direction	Protocol	Description
clk	slave	ClockSignal	Main PrimeCell SSP clock input.
clkkin	slave	ClockSignal	PrimeCell SSP clock input.
clkout	master	ClockSignal	Clock output.
intr	master	Signal	Interrupt signaling.
pvbus	slave	PVBus	Slave port for connection to PV bus master/decoder.
rorintr	master	Signal	Receive overrun interrupt.
rtintr	master	Signal	Receive timeout interrupt. We don't implement time out interrupt.
rx_dma_port	master	PL080_DMAC_DmaPortProtocol	PrimeCell SSP receive DMA port.

Port	Direction	Protocol	Description
rx_d	slave	<a href="#">Value</a>	PrimeCell SSP receive data.
rxintr	master	<a href="#">Signal</a>	Receive FIFO service request port.
tx_dma_port	master	<a href="#">PL080_DMAC_DmaPortProtocol</a>	PrimeCell SSP transmit DMA port.
tx_d	master	<a href="#">Value</a>	PrimeCell SSP transmit data.
txintr	master	<a href="#">Signal</a>	Transmit FIFO service request.

### Parameters for PL022\_SSP

This component does not have any parameters.

## 3.283 PL030\_RTC

Defined in `LISA/PL030_RTC.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### About PL030\_RTC

ARM PrimeCell Real Time Clock(PL030).

### Iris and MTI instances for PL030\_RTC

This model has the following Iris instances:

Name	Instance type
PL030_RTC	<a href="#">PL030_RTC</a>
PL030_RTC.busslave	<a href="#">PVBUSlave</a>
PL030_RTC.timer	<a href="#">ClockTimerThread</a>
PL030_RTC.timer.timer	<a href="#">ClockTimerThread64</a>
PL030_RTC.timer.timer.thread	<a href="#">SchedulerThread</a>
PL030_RTC.timer.timer.thread_event	<a href="#">SchedulerThreadEvent</a>

This model has the following MTI trace components:

Name	Component type
PL030_RTC.busslave	<a href="#">PVBUSlave</a>

### Ports for PL030\_RTC

Port	Direction	Protocol	Description
clock	slave	<a href="#">ClockSignal</a>	Clock input, typically 1MHz, driving master count rate.



Port	Direction	Protocol	Description
intr	master	Signal	Interrupt signaling.
pvbuss	slave	PVBus	Slave port for connection to PV bus master/decoder.

### Parameters for PL030\_RTC

This component does not have any parameters.

## 3.284 PL031\_RTC

Defined in LISA/PL031\_RTC.lisa.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r1p3	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### About PL031\_RTC

This component can provide a basic alarm function or long time base counter.

It has no impact on the performance of a PV system when idle or counting down. The component only executes code when the counter expires or during bus accesses

### Iris and MTI instances for PL031\_RTC

This model has the following Iris instances:

Name	Instance type
PL031_RTC	PL031_RTC
PL031_RTC.busslave	PVBusSlave
PL031_RTC.timer	ClockTimerThread
PL031_RTC.timer.timer	ClockTimerThread64
PL031_RTC.timer.timer.thread	SchedulerThread
PL031_RTC.timer.timer.thread_event	SchedulerThreadEvent

This model has the following MTI trace components:

Name	Component type
PL031_RTC.busslave	PVBusSlave

### Ports for PL031\_RTC

Port	Direction	Protocol	Description
clock	slave	ClockSignal	Clock input, typically 1MHz, driving master count rate.

Port	Direction	Protocol	Description
<code>intr</code>	master	Signal	Interrupt signaling.
<code>pvbuss</code>	slave	PVBus	Slave port for connection to PV bus master/decoder.

## Parameters for PL031\_RTC

### **RTCDR\_reset\_value**

Reset value for RTCDR.

Type: `uint32_t`

Default value: 0

### **RTCDR\_use\_current\_time**

Use current Unix/POSIX time for reset value for RTCDR. If true `RTCDR_reset_value` is ignored.

Type: `bool`

Default value: 1

## 3.285 PL041\_AACI

Defined in `LISA/PL041_AACI.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### About PL041\_AACI

The PL041\_AACI component is designed to connect to an audio output component such as [AudioOut\\_File](#) or [AudioOut\\_SDL](#).

The ability to play audio through this component depends on the AudioOut component in use and on the performance requirements of the software running on the simulated system. The rate of FIFO draining is controlled by the audio output to which the component is connected. This might not correspond to the rate that would be expected from the reference clock.

This component also contains a minimal register model of the LM4529 secondary codec as implemented on development boards supplied by Arm.

**Note**

This component is not a complete implementation of the AACI because the following functionality is not implemented:

- Audio input
- DMA access to FIFOs, rather than Programmed I/O
- Programming of the secondary codec through FIFOs rather than slot registers

## Iris and MTI instances for PL041\_AACI

This model has the following Iris instances:

Name	Instance type
PL041_AACI	PL041_AACI
PL041_AACI.busslave	PVBusSlave
PL041_AACI.timer	ClockTimerThread
PL041_AACI.timer.timer	ClockTimerThread64
PL041_AACI.timer.timer.thread	SchedulerThread
PL041_AACI.timer.timer.thread_event	SchedulerThreadEvent

This model has the following MTI trace components:

Name	Component type
PL041_AACI.busslave	PVBusSlave

## Ports for PL041\_AACI

Port	Direction	Protocol	Description
audio	master	AudioControl	Used to communicate with an audio out device.
clk_in_ref	slave	ClockSignal	Reference clock input, typically 25MH.
dma_rx	master	PL080_DMAC_DmaPortProtocol	DMA receive port.
dma_tx	master	PL080_DMAC_DmaPortProtocol	DMA transmit port.
irq	master	Signal	Single IRQ output port.
pvbuss	slave	PVBus	Slave port for connection to PV bus master/decoder.

## Parameters for PL041\_AACI

### **enabled**

Host interface connection enabled.

Type: `bool`

Default value: `true`

## 3.286 PL050\_KMI

Defined in `LISA/PL050_KMI.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### About PL050\_KMI

This model communicates with models of PS/2-like devices, for example a PS2Keyboard or PS2Mouse.

### Iris and MTI instances for PL050\_KMI

This model has the following Iris instances:

Name	Instance type
PL050_KMI	PL050_KMI
PL050_KMI.busslave	PVBusSlave
PL050_KMI.clk_divider	ClockDivider

This model has the following MTI trace components:

Name	Component type
PL050_KMI.busslave	PVBusSlave
PL050_KMI.clk_divider	ClockDivider

### Ports for PL050\_KMI

Port	Direction	Protocol	Description
clock	slave	ClockSignal	Clock input, typically 1MHz, which sets the master transmit/receive rate.
intr	master	Signal	Master port signaling completion of transmit or receive.
ps2device	slave	PS2Data	Used to communicate with a PS/2-like device.
pvbus	slave	PVBus	Slave port for connection to PV bus master/decoder.

### Parameters for PL050\_KMI

#### `clk_divider.div`

Clock Rate Divider.

Type: `uint64_t`

Default value: 1

**clk\_divider.mul**

Clock Rate Multiplier.

Type: uint64\_t

Default value: 1

## 3.287 PL061\_GPIO

Defined in LISA/PL061\_GPIO.lisa.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### About PL061\_GPIO

This component provides eight programmable inputs or outputs. Ports of different widths can be created by multiple instantiation. In addition, an interrupt interface is provided to configure any number of pins as interrupt sources.

### Iris and MTI instances for PL061\_GPIO

This model has the following Iris instances:

Name	Instance type
PL061_GPIO	PL061_GPIO
PL061_GPIO.busslave	PVBusSlave

This model has the following MTI trace components:

Name	Component type
PL061_GPIO.busslave	PVBusSlave

### Ports for PL061\_GPIO

Port	Direction	Protocol	Description
GPIO_In	slave	Value	Input lines. 32-bit data in, only [7:0] is used.
GPIO_Intr	master	Signal	Interrupt signal indicating to an interrupt controller that an interrupt occurred in one or more of the GPIO_In lines.
GPIO_MIS	master	Value	Indicates the masked interrupt status. 32-bit data out , only [7:0] is used. NOT necessary, as the GPIOMIS can be read from address 0x418.
GPIO_Out	master	Value	Output lines. 32-bit data out, only [7:0] is used.
pvbuss	slave	PVBus	Subordinate port for register access.

## Parameters for PL061\_GPIO

### **init\_inputs**

Parameter to set default values for inputs – reset only.

Type: uint32\_t

Default value: 0x0

## 3.288 PL080\_DMAC

Defined in LISA/PL080\_DMAC.lisa.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r1p3	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### About PL080\_DMAC

This component provides 8 configurable DMA channels and 16 DMA ports for handshaking with peripherals. You can configure each channel to operate in one of eight flow control modes either under DMA control or the control of the source or destination peripheral. Transfers can occur on either master channel and can optionally be endian-converted on both source and destination transfers.

This component might have a significant impact on system performance in certain flow control modes. Channels configured for small bursts, or using single bursts, and with peripheral DMA handshaking could add significant overhead. The peripheral has not been fully optimized to make use of the advanced features of the PVBUS model.

### Iris and MTI instances for PL080\_DMAC

This model has the following Iris instances:

Name	Instance type
PL080_DMAC	PL080_DMAC
PL080_DMAC.busmasterY (where Y = 0-1)	PVBusMaster
PL080_DMAC.busslave	PVBusSlave
PL080_DMAC.timer	ClockTimerThread
PL080_DMAC.timer.timer	ClockTimerThread64
PL080_DMAC.timer.timer.thread	SchedulerThread
PL080_DMAC.timer.timer.thread_event	SchedulerThreadEvent

This model has the following MTI trace components:

Name	Component type
PL080_DMAC.busmasterY (where Y = 0–1)	PVBusMaster
PL080_DMAC.busslave	PVBusSlave

### Ports for PL080\_DMAC

Port	Direction	Protocol	Description
clk_in	slave	ClockSignal	Clock signal to control DMA transfer rate.
dma_port	slave	PL080_DMAC_DmaPortProtocol	request/response ports for communicating with devices.
interr	master	Signal	DMA error interrupt signal.
intr	master	Signal	Combined DMA error and terminal count signal.
inttc	master	Signal	DMA terminal count signal.
pvb0_m	master	PVBus	Master bus interface 0 for DMA transfers.
pvb1_m	master	PVBus	Master bus interface 1 for DMA transfers.
pvb_s	slave	PVBus	Slave port for register accesses.
reset_in	slave	Signal	System reset.

### Parameters for PL080\_DMAC

#### **activate\_delay**

request delay.

Type: uint32\_t

Default value: 0

#### **fifo\_size**

Channel FIFO size in bytes.

Type: uint32\_t

Default value: 16

#### **generate\_clear**

Generate clear response.

Type: bool

Default value: false

#### **max\_transfer**

Largest atomic transfer.

Type: uint32\_t

Default value: 256

## 3.289 PL110\_CLCD

Defined in `LISA/PL110_CLCD.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r1p2	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### About PL110\_CLCD

This implementation provides a register model of the LCD controller.

You can connect the model through a framebuffer port to a visualization component, for example, so that LCD output can be viewed.

The implementation is optimized for situations where the majority of the framebuffer does not change. For instance, displaying full-screen video results in significantly reduced performance. Rendering pixel data into an appropriate form for the framebuffer port (rasterization) can also take a significant amount of simulation time. If the pixel data are coming from a PVBUSSlave region that has been configured as memory-like, rasterization only occurs in regions where memory contents are modified.

### Iris and MTI instances for PL110\_CLCD

This model has the following Iris instances:

Name	Instance type
PL110_CLCD	PL110_CLCD
PL110_CLCD.pl11x_clcd	PL11x_CLCD
PL110_CLCD.pl11x_clcd.busmaster	PVBusMaster
PL110_CLCD.pl11x_clcd.busslave	PVBusSlave
PL110_CLCD.pl11x_clcd.timer	ClockTimerThread
PL110_CLCD.pl11x_clcd.timer.timer	ClockTimerThread64
PL110_CLCD.pl11x_clcd.timer.timer.thread	SchedulerThread
PL110_CLCD.pl11x_clcd.timer.timer.thread_event	SchedulerThreadEvent

This model has the following MTI trace components:

Name	Component type
PL110_CLCD.pl11x_clcd.busmaster	PVBusMaster
PL110_CLCD.pl11x_clcd.busslave	PVBusSlave



## Ports for PL110\_CLCD

Port	Direction	Protocol	Description
clk_in	slave	ClockSignal	Master clock input, typically 24MHz, to drive pixel clock timing.
control	slave	Value	Auxiliary control register 1.
display	master	LCD	Connection to visualization component.
intr	master	Signal	Interrupt signaling for flyback events.
pvbush_m	master	PVBus	DMA port for video data.
pvbush	slave	PVBus	Slave port for register access.

## Parameters for PL110\_CLCD

### **disable\_snooping\_dma**

Disable DMA snooping.

Type: `bool`

Default value: `false`

### **pixel\_double\_limit**

Minimum LCD pixel width before display will be zoomed.

Type: `uint32_t`

Default value: `300`

## 3.290 PL111\_CLCD

Defined in `LISA/PL111_CLCD.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p2	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### About PL111\_CLCD

This component implements the hardware cursor support of the PL111\_CLCD, which is the main difference with PL110\_CLCD.

### Iris and MTI instances for PL111\_CLCD

This model has the following Iris instances:

Name	Instance type
PL111_CLCD	PL111_CLCD
PL111_CLCD.pl11x_clcd	PL11x_CLCD
PL111_CLCD.pl11x_clcd.busmaster	PVBusMaster
PL111_CLCD.pl11x_clcd.busslave	PVBusSlave
PL111_CLCD.pl11x_clcd.timer	ClockTimerThread
PL111_CLCD.pl11x_clcd.timer.timer	ClockTimerThread64
PL111_CLCD.pl11x_clcd.timer.timer.thread	SchedulerThread
PL111_CLCD.pl11x_clcd.timer.timer.thread_event	SchedulerThreadEvent

This model has the following MTI trace components:

Name	Component type
PL111_CLCD.pl11x_clcd.busmaster	PVBusMaster
PL111_CLCD.pl11x_clcd.busslave	PVBusSlave

### Ports for PL111\_CLCD

Port	Direction	Protocol	Description
clk_in	slave	ClockSignal	Master clock input, typically 24MHz, to drive pixel clock timing.
control	slave	Value	Auxiliary control register 1.
display	master	LCD	Connection to visualization component.
intr	master	Signal	Interrupt signaling for flyback events.
pvbus_m	master	PVBus	DMA port for video data.
pvbus	slave	PVBus	Slave port for register access.

### Parameters for PL111\_CLCD

#### **disable\_snooping\_dma**

Disable DMA snooping.

Type: `bool`

Default value: `false`

#### **pixel\_double\_limit**

Minimum LCD pixel width before display will be zoomed.

Type: `uint32_t`

Default value: `300`

## 3.291 PL180\_MCI

Defined in `LISA/PL180_MCI.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### About PL180\_MCI

When paired with an MMC card model, the PL180\_MCI component provides emulation of a flexible, persistent storage mechanism. See [MMC](#). The PL180\_MCI component fully models the registers of the corresponding PrimeCell, but supports a subset of the functionality of the PL180:

- The controller supports block mode transfers, but does not currently support streaming data transfer.
- The controller can be attached to a single MMC device. The MMC bus mode and SDIO modes of the PL180 PrimeCell are not supported.
- Command and Data timeouts are not simulated.
- Payload CRC errors are not simulated.
- The DMA interface present in the PL180 PrimeCell is not modeled.
- Minimal timing is implemented within the model.



Note

At compile time, you can enable command tracing within the PL180\_MCI component by modifying the `PL180_TRACE` macro in the `MMC.lisa` file. This sends command and event trace to standard output. You can use this output to help diagnose device driver and controller-to-card protocol issues.

### Iris and MTI instances for PL180\_MCI

This model has the following Iris instances:

Name	Instance type
PL180_MCI	<a href="#">PL180_MCI</a>
PL180_MCI.busslave	<a href="#">PVBusSlave</a>

This model has the following MTI trace components:

Name	Component type
PL180_MCI.busslave	<a href="#">PVBusSlave</a>

## Ports for PL180\_MCI

Port	Direction	Protocol	Description
MCIINTR	master	Signal	-
mmc_m	master	MMC_Protocol	-
pvbuss	slave	PVBus	-

## Parameters for PL180\_MCI

### pl180\_fifo\_depth

PL180 FIFO Depth.

Type: int

Default value: 16

## 3.292 PL192\_VIC

Defined in LISA/PL192\_VIC.lisa.

This model supports the following revisions of the IP at the given quality levels:

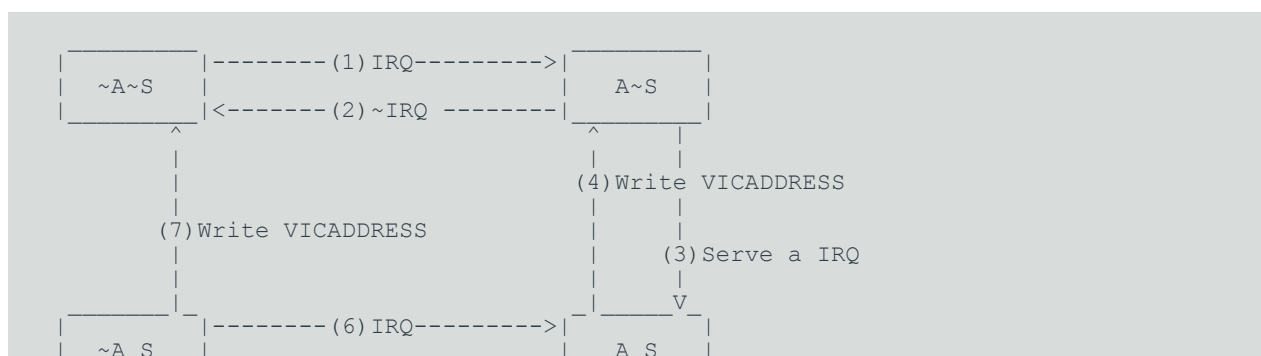
Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### About PL192\_VIC

This component aggregates interrupts and generates interrupt signals for the Arm processor. When coupled with an Arm processor that provides a VIC port, routing to the appropriate interrupt handler can optionally be performed in hardware, reducing interrupt latency. The PL192\_VIC can also be daisy-chained with other PL192 VICs to permit more than 32 interrupts. The VIC supports hardware and software prioritization of interrupts.

This is the state transition diagram of a VIC 192 interrupt source:



| \_\_\_\_\_ | <----- (5) ~IRQ----- | \_\_\_\_\_ |

**A**

The IRQ is active. It is in the irqServeList which is a sorted list of active IRQs that need to be served.

**~A**

Inactive IRQ. The corresponding input port is tied to low.

**S**

The IRQ is being served.

**~S**

The IRQ is waiting to be served

## Detailed Descriptions

**1**

An IRQ is asserted, and state changes from ~A~S to A~S. The IRQ is inserted into a sorted list called irqServeList to wait for service. In this case, the corresponding input pin is tied to high.

**2**

The IRQ is deasserted. This can happen when the device does not want to keep IRQ active. For example, after continuously sending data, a UART can deassert IRQ to indicate stopping data transmission. In this case, even if the IRQ is in the stack and ready to be served it should be removed from the stack immediately.

**3**

When an IRQ that is at the top of the waiting stack is being served, the state changes from A~S to AS.

**4**

Writing to the VICADDRESS register indicates that the current served interrupt has been finished. However, the state of the IRQ could be still active. The device, such as a UART, that raised the IRQ could still want to generate a new IRQ to finish a task.

**5**

The IRQ is being served. Before finishing, the device deasserts the IRQ. The current IRQ will be removed from the top of the stack immediately, but it is still being served.

**6**

The IRQ is being served, and it is not in the stack. At this point, the device reasserts the IRQ and the state of IRQ changes from ~AS to AS. In another words, the IRQ is reinserted into the stack.

**7**

An ISR writes to VICADDRESS to indicate the current IRQ has been served. Meanwhile the IRQ is deasserted by the device, the state of the VIC changes from ~A S to ~A~S.

The handshake when VIC is using VIC port to communicate. As this is an untimed model, it is not possible to model the timed nature of vector address passing accurately. There are two options offered:

1. Send the address just after the IRQ. This is closer to the hardware but requires that daisy-chained VICs repeatedly send their address as new, higher priority IRQs arrive, so may be slower:



2. Send the address during the ack. In this case, the ack ripples up through the VICs until it finds the IRQ and then the address ripples back down through the VICs, before the ack returns:



In both cases the ack clear is ignored by the VIC.

## Iris and MTI instances for PL192\_VIC

This model has the following Iris instances:

Name	Instance type
PL192_VIC	PL192_VIC
PL192_VIC.busslave	PVBusSlave

This model has the following MTI trace components:

Name	Component type
PL192_VIC.busslave	PVBusSlave

## Ports for PL192\_VIC

Port	Direction	Protocol	Description
nVICFIQ	master	Signal	Send out FIQ signal to the next level VIC or CPI.
nVICFIQIN	slave	Signal	Used to receive FIQ signal when daisy chained.
nVICIRQ	master	Signal	Send out IRQ signal to the next level VIC or processor.
nVICIRQIN	slave	Signal	Used to receive IRQ signal when daisy chained.
pvbus	slave	PVBus	Slave port for register access.
VICIntSource	slave	Signal	Interrupt source input sources.

Port	Direction	Protocol	Description
VICIRQACK	slave	Signal	Receive acknowledge signal from next level VIC or processor.
VICIRQACKOUT	master	Signal	Used to send out acknowledge signals when daisy chained.
VICVECTADDRIN	slave	ValueState	Used to receive vector address when daisy chained.
VICVECTADDRROUT	master	ValueState	Used to send vector address to next level VIC or processor.

### Parameters for PL192\_VIC

This component does not have any parameters.

## 3.293 PL310\_L2CC

Defined in `LISA/PL310_L2CC.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### About PL310\_L2CC

The presence of additional on-chip secondary cache can improve performance when significant memory traffic is generated by the processor. A secondary cache assumes the existence of a Level 1, or primary, cache that is closely coupled or internal to the processor.

This component has two modes of operation, which are controlled by the `cache-state_modelled` parameter:

#### Register view

Cache control registers are present but the cache behavior is not modeled.

#### Functional model

Cache behavior is modeled.

Arm supports the use of the PL310 when connected to the Arm® Cortex®-A5 or Cortex-A9 processor.

This component implements the programmer-visible functionality of the PL310, and excludes some non-programmer visible features. The following features are implemented in the model:

- Physically addressed and physically tagged.
- Lockdown format C supported, for data and instructions. Lockdown format C is also known as way locking.
- Lockdown by line supported.
- Lockdown by manager ID supported.

- Direct mapped to 16-way associativity, depending on the configuration and the use of lockdown registers. The associativity is configurable as 8 or 16.
- L2 cache available size can be 16 KB to 8 MB, depending on configuration and the use of the lockdown registers.
- Fixed line length of 32 bytes (8 words or 256 bits).
- Supports all of the AXI cache modes:
  - Write-through and write-back.
  - Read allocate, write allocate, read and write allocate.
- Force write-allocate option to always have cacheable writes allocated to L2 cache, for processors not supporting this mode.
- Normal memory non-cacheable shared reads are treated as cacheable non-allocatable. Normal memory non-cacheable shared writes are treated as cacheable write-through no write-allocate. There is an option, Shared Override, to override this behavior.
- TrustZone support, with the following features:
  - Non-Secure (NS) tag bit added in tag RAM and used for lookup in the same way as an address bit.
  - NS bit in Tag RAM used to determine security level of evictions to L3.
  - Restrictions for NS accesses for control, configuration, and maintenance registers to restrict access to secure data.
- Pseudo-Random victim selection policy. You can make this deterministic by using lockdown registers.
- Software option to enable exclusive cache configuration.
- Configuration registers accessible using address decoding in the component.
- Interrupt triggering in case of an error response when accessing L3.
- Maintenance operations.
- Prefetching capability.

The performance of this component depends on the configuration of the associated L1 caches and the mode it is in:

#### **Register mode**

No significant effect.

#### **Functional mode with functional-mode L1**

The addition of a functional L2 cache has minimal further impact on performance when running applications that are cache-bound.

#### **Functional mode with a register-mode L1**

There is a significant impact on system performance.





Setting timing delays in this model does not impact the simulation speed. Generally, timing delays are only modeled for CPUs.

## Differences between the model and the RTL

This model does not implement the following features, most of which are not relevant from a PV modeling point of view:

- There is no interface to the data and tag RAM as they are embedded in the model.
- Critical word first linefill is not supported, as it is not relevant for PV modeling.
- Buffers are not modeled.
- Outstanding accesses on slave and master ports cannot occur by design in a PV model as all transactions are atomic.
- Option to select one or two master ports and option to select one or two slave ports is not supported. Only one master port and one slave port are supported.
- Clock management and power modes are not supported, as they are not relevant for PV modeling.
- Wait, latency, clock enable, parity, and error support for data and tag RAMs are not included, as they are not relevant for PV modeling, and the data and tag RAMs embedded in the model cannot generate error responses.
- MBIST support is not included.
- Debug mode and debug registers are not supported.
- Test mode and scan chains are not supported.
- L2 cache event monitoring is not supported.
- Address filtering in the master ports is not supported.
- Performance counters are not supported.
- These Cortex-A9-related optimizations are not supported:
  - Prefetch hints
  - Full line of zero
  - Early write response
- Hazard detection is not required because of the atomic nature of the accesses in PV modeling and the fact that buffers are not modeled, therefore hazards cannot occur.
- Registers that belong to unimplemented features are accessible but do not have any functionality.

This model implements the following features differently to the hardware:

- Error handling. DECERR from the master port is mapped to SLVERR. Internal errors in cache RAM, for example parity errors, cannot happen in the model.
- Background cache operations do not occur in the background. They occur atomically.

- The LOCKDOWN\_BY\_LINE and LOCKDOWN\_BY\_MASTER parameter values are reflected in the CacheType register, but the feature is not switched off when the parameter is 0.
- This feature is additional:
  - Data RAM and Tag RAM are embedded in the model.

### Iris and MTI instances for PL310\_L2CC

This model has the following Iris instances:

Name	Instance type
PL310_L2CC	PL310_L2CC

This model has the following MTI trace components:

Name	Component type
PL310_L2CC	PL310_L2CC

### Ports for PL310\_L2CC

Port	Direction	Protocol	Description
DECERRINTR	master	Signal	Decode error received on master port from L3.
ECNTRINTR	master	Signal	Event Counter Overflow / Increment.
ERRRDINTR	master	Signal	Error on L2 data RAM read.
ERRRTINTR	master	Signal	Error on L2 tag RAM read.
ERRWDINTR	master	Signal	Error on L2 data RAM write.
ERRWTINTR	master	Signal	Error on L2 tag RAM write.
L2CCINTR	master	Signal	Combined interrupt output.
PARRDINTR	master	Signal	Parity error on L2 data RAM read.
PARRTINTR	master	Signal	Parity error on L2 tag RAM read.
pvbus_m	master	PVBus	Master port for connection to PV bus master/decoder.
pvbus_s	slave	PVBus	Slave port for connection to PV bus master/decoder.
SLVERRINTR	master	Signal	Slave error on master port from L3.

### Parameters for PL310\_L2CC

#### ASSOCIATIVITY

Associativity for Auxiliary Control Register.

Type: uint32\_t

Default value: 0

#### CACHEID

Cache controller cache ID.

Type: uint32\_t

Default value: 0

**CFGBIGEND**

Big-endian mode for accessing configuration registers out of reset.

Type: `uint32_t`

Default value: 0

**LOCKDOWN\_BY\_LINE**

Lockdown by line - value is reflected in CacheType register Bit 25, but the feature is not switched off when the parameter is 0.

Type: `uint32_t`

Default value: 0

**LOCKDOWN\_BY\_MASTER**

Lockdown by master - value is reflected in CacheType register Bit 26, but the feature is not switched off when the parameter is 0.

Type: `uint32_t`

Default value: 0

**REGFILEBASE**

Base address for accessing configuration registers.

Type: `uint32_t`

Default value: `0x1f002000`

**WAYSIZE**

Size of ways for Auxiliary Control Register.

Type: `uint32_t`

Default value: 1

**cache-state\_modelled**

Specifies whether real cache state is modelled (vs. register model).

Type: `bool`

Default value: `false`

**delay\_cache\_hit**

Cost to handle a cache hit.

Type: uint32\_t

Default value: 0

**delay\_cache\_miss**

Cost to handle a cache miss.

Type: uint32\_t

Default value: 0

**delay\_cache\_perbeat**

Cost to handle one beat of cache data movement.

Type: uint32\_t

Default value: 0

3.294 PL330\_DMAC

Defined in LISA/PL330\_DMAC.lisa.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

About PL330\_DMAC

The model uses a single LISA+ component but with a C++ model for each of the channels included in the LISA+ file. Enabled channels are kept on an enabled channels stack in priority order. When a channel state changes, re-arbitration takes place to make the highest (topmost) channel active.

Each transaction carries the identity of the requesting thread. This controller has up to eight channel threads and a manager thread. Each has an ID. In the hardware:

ID

AxID[3:0]

Identifying channels

0x0 - (numberOfChannels - 1)

## Managers

numberOfChannels

For example, 0x0-0x7 and 0x8, respectively. The manager originates only instruction fetches, and the manager ID is also used for instruction fetches issued by the channels.

In the model, the identity of the requesting thread is encoded into each transaction using the low-order 16 bits of the Manager ID field:

- Channel data: 0-7.
- Channel instruction fetch: 0xffff.
- Manager instruction fetch: 0xffff.

If a downstream component needs to know the IDs of bus masters that use either the low-order 16 bits or the label, use the label. The LabellerForDMA330 component shifts the low-order 16 bits into the label, while providing a degree of control over the label encoding. The example below maintains separate IDs for each data channel while using the correct hardware ID to identify instruction fetch for a DMA-330 with 8 channels:

```
pl330_dma : PL330_DMAC( "p_max_channels" = 8 );
dma_labeller : LabellerForDMA330(
    "dma330_discriminate_data_channels" = true,
    "dma330_s_instruction_label" = 8,
    "dma330_ns_instruction_label" = 8 );
pl330_dma.pvbus_m => dma_labeller.pvbus_s;
dma_labeller.pvbus_m => output_bus.pvbus_s;
```

## Iris and MTI instances for PL330\_DMAC

This model has the following Iris instances:

Name	Instance type
PL330_DMAC	PL330_DMAC
PL330_DMAC.busmaster	PVBusMaster
PL330_DMAC.busslave	PVBusSlave
PL330_DMAC.busslave_ns	PVBusSlave
PL330_DMAC.timer	ClockTimerThread
PL330_DMAC.timer.timer	ClockTimerThread64
PL330_DMAC.timer.timer.thread	SchedulerThread
PL330_DMAC.timer.timer.thread_event	SchedulerThreadEvent

This model has the following MTI trace components:

Name	Component type
PL330_DMAC.busmaster	PVBusMaster
PL330_DMAC.busslave	PVBusSlave
PL330_DMAC.busslave_ns	PVBusSlave

## Ports for PL330\_DMAC

Port	Direction	Protocol	Description
clk_in	slave	ClockSignal	-
irq_abort_master_port	master	Signal	-
irq_master_port	master	Signal	-
pvbus_m	master	PVBus	-
pvbus_s_ns	slave	PVBus	-
pvbus_s	slave	PVBus	-
reset_in	slave	Signal	-

## Parameters for PL330\_DMAC

### **activate\_delay**

request delay.

Type: `uint32_t`

Default value: 0

### **fifo\_size**

Channel FIFO size in bytes.

Type: `uint32_t`

Default value: 16

### **generate\_clear**

Generate clear response.

Type: `bool`

Default value: `false`

### **max\_transfer**

Largest atomic transfer.

Type: `uint32_t`

Default value: 256

### **p\_axi\_bus\_width\_param**

AXI bus width.

Type: `uint32_t`

Default value: 32

**p\_buffer\_depth**

buffer depth.

Type: `uint32_t`

Default value: 16

**p\_cache\_line\_words**

number of words in a cache line.

Type: `uint32_t`

Default value: 1

**p\_cache\_lines**

number of cache lines.

Type: `uint32_t`

Default value: 1

**p\_controller\_boots**

DMA boots from reset.

Type: `bool`

Default value: `true`

**p\_controller\_nsecure**

Controller non-secure at reset (`boot_manager_ns`).

Type: `bool`

Default value: `false`

**p\_irq\_nsecure**

Interrupts non-secure at reset.

Type: `uint32_t`

Default value: 0x00000000

**p\_lsq\_read\_size**

LSQ read buffer depth.

Type: `uint32_t`

Default value: 4

**p\_lsq\_write\_size**

LSQ write buffer depth.

Type: uint32\_t

Default value: 4

**p\_max\_channels**

virtual channels.

Type: uint32\_t

Default value: 8

**p\_max\_irqs**

number of interrupts.

Type: uint32\_t

Default value: 32

**p\_max\_periph**

number of peripheral interfaces.

Type: uint32\_t

Default value: 32

**p\_perip\_request\_acceptance\_0**

Peripheral 0 request acceptance.

Type: uint32\_t

Default value: 2

**p\_perip\_request\_acceptance\_1**

Peripheral 1 request acceptance.

Type: uint32\_t

Default value: 2

**p\_perip\_request\_acceptance\_10**

Peripheral 10 request acceptance.

Type: uint32\_t

Default value: 2



**p\_perip\_request\_acceptance\_11**

Peripheral 11 request acceptance.

Type: uint32\_t

Default value: 2

**p\_perip\_request\_acceptance\_12**

Peripheral 12 request acceptance.

Type: uint32\_t

Default value: 2

**p\_perip\_request\_acceptance\_13**

Peripheral 13 request acceptance.

Type: uint32\_t

Default value: 2

**p\_perip\_request\_acceptance\_14**

Peripheral 14 request acceptance.

Type: uint32\_t

Default value: 2

**p\_perip\_request\_acceptance\_15**

Peripheral 15 request acceptance.

Type: uint32\_t

Default value: 2

**p\_perip\_request\_acceptance\_16**

Peripheral 16 request acceptance.

Type: uint32\_t

Default value: 2

**p\_perip\_request\_acceptance\_17**

Peripheral 17 request acceptance.

Type: uint32\_t

Default value: 2

**p\_perip\_request\_acceptance\_18**

Peripheral 18 request acceptance.

Type: uint32\_t

Default value: 2

**p\_perip\_request\_acceptance\_19**

Peripheral 19 request acceptance.

Type: uint32\_t

Default value: 2

**p\_perip\_request\_acceptance\_2**

Peripheral 2 request acceptance.

Type: uint32\_t

Default value: 2

**p\_perip\_request\_acceptance\_20**

Peripheral 20 request acceptance.

Type: uint32\_t

Default value: 2

**p\_perip\_request\_acceptance\_21**

Peripheral 21 request acceptance.

Type: uint32\_t

Default value: 2

**p\_perip\_request\_acceptance\_22**

Peripheral 22 request acceptance.

Type: uint32\_t

Default value: 2

**p\_perip\_request\_acceptance\_23**

Peripheral 23 request acceptance.

Type: uint32\_t

Default value: 2

**p\_perip\_request\_acceptance\_24**

Peripheral 24 request acceptance.

Type: uint32\_t

Default value: 2

**p\_perip\_request\_acceptance\_25**

Peripheral 25 request acceptance.

Type: uint32\_t

Default value: 2

**p\_perip\_request\_acceptance\_26**

Peripheral 26 request acceptance.

Type: uint32\_t

Default value: 2

**p\_perip\_request\_acceptance\_27**

Peripheral 27 request acceptance.

Type: uint32\_t

Default value: 2

**p\_perip\_request\_acceptance\_28**

Peripheral 28 request acceptance.

Type: uint32\_t

Default value: 2

**p\_perip\_request\_acceptance\_29**

Peripheral 29 request acceptance.

Type: uint32\_t

Default value: 2

**p\_perip\_request\_acceptance\_3**

Peripheral 3 request acceptance.

Type: uint32\_t

Default value: 2

**p\_perip\_request\_acceptance\_30**

Peripheral 30 request acceptance.

Type: uint32\_t

Default value: 2

**p\_perip\_request\_acceptance\_31**

Peripheral 31 request acceptance.

Type: uint32\_t

Default value: 2

**p\_perip\_request\_acceptance\_4**

Peripheral 4 request acceptance.

Type: uint32\_t

Default value: 2

**p\_perip\_request\_acceptance\_5**

Peripheral 5 request acceptance.

Type: uint32\_t

Default value: 2

**p\_perip\_request\_acceptance\_6**

Peripheral 6 request acceptance.

Type: uint32\_t

Default value: 2

**p\_perip\_request\_acceptance\_7**

Peripheral 7 request acceptance.

Type: uint32\_t

Default value: 2

**p\_perip\_request\_acceptance\_8**

Peripheral 8 request acceptance.

Type: uint32\_t

Default value: 2

**p\_perip\_request\_acceptance\_9**

Peripheral 9 request acceptance.

Type: `uint32_t`

Default value: 2

**p\_periph\_nsecure**

Peripherals non-secure at reset.

Type: `bool`

Default value: `false`

**p\_read\_issuing\_capability**

AXI read issuing capability.

Type: `uint32_t`

Default value: 1

**p\_reset\_pc**

DMA PC at reset.

Type: `uint32_t`

Default value: `0x60000000`

**p\_write\_issuing\_capability**

AXI write issuing capability.

Type: `uint32_t`

Default value: 1

**revision**

revision ID.

Type: `string`

Default value: `"r0p0"`

## 3.295 PL340\_DMC

Defined in `LISA/PL340_DMC.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### About PL340\_DMC

This component provides an interface for up to four DRAM chips. The implementation also provides an APB interface to configure the controller behavior. You can access the registers through the APB interface.

### Iris and MTI instances for PL340\_DMC

This model has the following Iris instances:

Name	Instance type
PL340_DMC	PL340_DMC
PL340_DMC.apb_slave	PVBusSlave
PL340_DMC.exclusive_monitorY (where Y = 0-3)	PVBusExclusiveMonitor
PL340_DMC.exclusive_monitorY.bus_mapper (where Y = 0-3)	PVBusMapper

This model has the following MTI trace components:

Name	Component type
PL340_DMC.apb_slave	PVBusSlave
PL340_DMC.exclusive_monitorY (where Y = 0-3)	PVBusExclusiveMonitor
PL340_DMC.exclusive_monitorY.bus_mapper (where Y = 0-3)	PVBusMapper

### Ports for PL340\_DMC

Port	Direction	Protocol	Description
apb_interface	slave	PVBus	Receive the apb config read/writes here.
axi_if_in	slave	PVBus	Receive the axi reads/writes here; up to four chips can be connected.
axi_if_out	master	PVBus	The output ports where the actual mem chips are connected.

### Parameters for PL340\_DMC

#### IF\_CHIPO

Set this parameter to 0 if memory is connected.

Type: int

Default value: -1

**IF\_CHIP1**

Set this parameter to 0 if memory is connected.

Type: `int`

Default value: -1

**IF\_CHIP2**

Set this parameter to 0 if memory is connected.

Type: `int`

Default value: -1

**IF\_CHIP3**

Set this parameter to 0 if memory is connected.

Type: `int`

Default value: -1

**MEMORY\_WIDTH**

Set this parameter to 0 if memory is connected.

Type: `int`

Default value: 32

3.296 PL350\_SMC

Defined in `LISA/PL350_SMC.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r1p2	Full support
r2p2	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

About PL350\_SMC

This component provides two memory interfaces. Each interface can be connected to a maximum of four memory devices, giving a total of eight inputs from the PVBUSDecoder and eight outputs

to either SRAM or NAND devices. Only one kind of memory can be connected to a particular interface, either SRAM or NAND.

It provides a PVBUS slave to control the device behavior. A remap port is also provided to assist in remapping particular memory regions.

This component is optimized to have negligible impact on transaction performance, except when memory remap settings are changed, when there might be a significant effect.

## Iris and MTI instances for PL350\_SMC

This model has the following Iris instances:

Name	Instance type
PL350_SMC	PL350_SMC
PL350_SMC.addr_remapper	TZSwitch
PL350_SMC.addr_remapper.pvbus_mapper	PVBusMapper
PL350_SMC.apb_slave	PVBusSlave
PL350_SMC.exclusive_monitorY_Z (where Y = 0-1; Z = 0-3)	PVBusExclusiveMonitor
PL350_SMC.exclusive_monitorY_Z.bus_mapper (where Y = 0-1; Z = 0-3)	PVBusMapper
PL350_SMC.master_ifY_Z (where Y = 0-1; Z = 0-3)	PVBusMaster
PL350_SMC.nand_remap_slave	PVBusSlave

This model has the following MTI trace components:

Name	Component type
PL350_SMC.addr_remapper.pvbus_mapper	PVBusMapper
PL350_SMC.apb_slave	PVBusSlave
PL350_SMC.exclusive_monitorY_Z (where Y = 0-1; Z = 0-3)	PVBusExclusiveMonitor
PL350_SMC.exclusive_monitorY_Z.bus_mapper (where Y = 0-1; Z = 0-3)	PVBusMapper
PL350_SMC.master_ifY_Z (where Y = 0-1; Z = 0-3)	PVBusMaster
PL350_SMC.nand_remap_slave	PVBusSlave

## Ports for PL350\_SMC

Port	Direction	Protocol	Description
apb_interface	slave	PVBus	This is where we expect to receive all the APB data which is used to read/write the device regs.
axi_chip_if0_in	slave	PVBus	This is where we expect to receive all the AXI data which is used to read/wrie NAND/RAM mem.
axi_chip_if0_out	master	PVBus	Manager interface 0 to connect to SRAM/NAND.
axi_chip_if1_in	slave	PVBus	This is where we expect to receive all the AXI data which is used to read/wrie NAND/RAM mem.
axi_chip_if1_out	master	PVBus	Manager interface 1 to connect to SRAM/NAND.
axi_remap	slave	PVBus	This is the remap port that the designer needs to connect to zero.
irq_in_if0	slave	Signal	-



Port	Direction	Protocol	Description
irq_in_if1	slave	Signal	-
irq_out	master	Signal	Interrupt port.
nand_remap_port	slave	PVBus	-

## Parameters for PL350\_SMC

### **IF0\_CHIP0**

Interface 0 chip 0 connected.

Type: `bool`

Default value: `false`

### **IF0\_CHIP0\_BASE**

Interface 0 chip 0 Base address.

Type: `int`

Default value: 0

### **IF0\_CHIP0\_SIZE**

Interface 0 chip 0 Size.

Type: `int`

Default value: 0

### **IF0\_CHIP1**

Interface 0 chip 1 connected.

Type: `bool`

Default value: `false`

### **IF0\_CHIP1\_BASE**

Interface 0 chip 1 Base address.

Type: `int`

Default value: 0

### **IF0\_CHIP1\_SIZE**

Interface 0 chip 1 Size.

Type: `int`

Default value: 0

**IFO\_CHIP2**

Interface 0 chip 2 connected.

Type: `bool`

Default value: `false`

**IFO\_CHIP2\_BASE**

Interface 0 chip 2 Base address.

Type: `int`

Default value: 0

**IFO\_CHIP2\_SIZE**

Interface 0 chip 2 Size.

Type: `int`

Default value: 0

**IFO\_CHIP3**

Interface 0 chip 3 connected.

Type: `bool`

Default value: `false`

**IFO\_CHIP3\_BASE**

Interface 0 chip 3 Base address.

Type: `int`

Default value: 0

**IFO\_CHIP3\_SIZE**

Interface 0 chip 3 Size.

Type: `int`

Default value: 0

**IFO\_MEM\_TYPE\_PARAMETER**

Interface 0 Mem type.

Type: `uint32_t`

Default value: 0

**IF1\_CHIP0**

Interface 1 chip 0 connected.

Type: `bool`

Default value: `false`

**IF1\_CHIP0\_BASE**

Interface 1 chip 0 Base address.

Type: `int`

Default value: 0

**IF1\_CHIP0\_SIZE**

Interface 1 chip 0 Size.

Type: `int`

Default value: 0

**IF1\_CHIP1**

Interface 1 chip 1 connected.

Type: `bool`

Default value: `false`

**IF1\_CHIP1\_BASE**

Interface 1 chip 1 Base address.

Type: `int`

Default value: 0

**IF1\_CHIP1\_SIZE**

Interface 1 chip 1 Size.

Type: `int`

Default value: 0

**IF1\_CHIP2**

Interface 1 chip 2 connected.

Type: `bool`

Default value: `false`

**IF1\_CHIP2\_BASE**

Interface 1 chip 2 Base address.

Type: `int`

Default value: 0

**IF1\_CHIP2\_SIZE**

Interface 1 chip 2 Size.

Type: `int`

Default value: 0

**IF1\_CHIP3**

Interface 1 chip 3 connected.

Type: `bool`

Default value: `false`

**IF1\_CHIP3\_BASE**

Interface 1 chip 3 Base address.

Type: `int`

Default value: 0

**IF1\_CHIP3\_SIZE**

Interface 1 chip 3 Size.

Type: `int`

Default value: 0

**IF1\_MEM\_TYPE\_PARAMETER**

Interface 1 Mem type.

Type: `uint32_t`

Default value: 0

**PERIPH\_ID\_0**

Periph\_ID\_0 value.

Type: `int`

Default value: 0x52

**REMAP**

Remap the device.

Type: `int`

Default value: -1

**addr\_remapper.normal**

Normal Port.

Type: `uint32_t`

Default value: 2

**addr\_remapper.secure**

Secure Port.

Type: `uint32_t`

Default value: 1

**exclusive\_monitor0\_0.apply\_access\_width\_criteria\_to\_non\_excl\_stores**

Apply the given exclusive store width matching criteria to non-exclusive stores.

Type: `bool`

Default value: `true`

**exclusive\_monitor0\_0.clear\_on\_strex\_address\_mismatch**

Whether monitor is cleared when strex fails due to address mismatch.

Type: `bool`

Default value: `true`

**exclusive\_monitor0\_0.enable\_component**

Enable component.

Type: `bool`

Default value: `true`

**exclusive\_monitor0\_0.exclusive\_monitor\_clear\_on\_atomic\_from\_same\_master**

Monitor atomics from the same master.

Type: `bool`

Default value: `true`

**`exclusive_monitor0_0.log2_granule_size`**

log2 of address granule size.

Type: `uint32_t`

Default value: 3

**`exclusive_monitor0_0.match_access_width`**

Fail STREX if not the same access width as LDREX.

Type: `bool`

Default value: `true`

**`exclusive_monitor0_0.match_secure_state`**

Treat the secure state like an address bit.

Type: `bool`

Default value: `true`

**`exclusive_monitor0_0.monitor_access_level`**

0: Monitor all accesses, 1: Monitor all accesses except WriteBack, 2: Only monitor accesses with memory type NonCacheable or Device.

Type: `uint32_t`

Default value: 0

**`exclusive_monitor0_0.monitor_non_excl_stores`**

Monitor non-exclusive stores from the same master.

Type: `bool`

Default value: `false`

**`exclusive_monitor0_0.number_of_monitors`**

Number of monitors.

Type: `unsigned`

Default value: 8

**exclusive\_monitor0\_0.shareability\_domain**

Maximum shareability domain of interest, transactions outside of the domain will pass through un-monitored (0-non-shared, 1-inner, 2-outer, 3-system).

Type: unsigned

Default value: 3

**exclusive\_monitor0\_1.apply\_access\_width\_criteria\_to\_non\_excl\_stores**

Apply the given exclusive store width matching criteria to non-exclusive stores.

Type: bool

Default value: true

**exclusive\_monitor0\_1.clear\_on\_strex\_address\_mismatch**

Whether monitor is cleared when strex fails due to address mismatch.

Type: bool

Default value: true

**exclusive\_monitor0\_1.enable\_component**

Enable component.

Type: bool

Default value: true

**exclusive\_monitor0\_1.exclusive\_monitor\_clear\_on\_atomic\_from\_same\_master**

Monitor atomics from the same master.

Type: bool

Default value: true

**exclusive\_monitor0\_1.log2\_granule\_size**

log2 of address granule size.

Type: uint32\_t

Default value: 3

**exclusive\_monitor0\_1.match\_access\_width**

Fail STREX if not the same access width as LDREX.

Type: bool

Default value: `true`

**exclusive\_monitor0\_1.match\_secure\_state**

Treat the secure state like an address bit.

Type: `bool`

Default value: `true`

**exclusive\_monitor0\_1.monitor\_access\_level**

0: Monitor all accesses, 1: Monitor all accesses except WriteBack, 2: Only monitor accesses with memory type NonCacheable or Device.

Type: `uint32_t`

Default value: 0

**exclusive\_monitor0\_1.monitor\_non\_excl\_stores**

Monitor non-exclusive stores from the same master.

Type: `bool`

Default value: `false`

**exclusive\_monitor0\_1.number\_of\_monitors**

Number of monitors.

Type: `unsigned`

Default value: 8

**exclusive\_monitor0\_1.shareability\_domain**

Maximum shareability domain of interest, transactions outside of the domain will pass through un-monitored (0-non-shared, 1-inner, 2-outer, 3-system).

Type: `unsigned`

Default value: 3

**exclusive\_monitor0\_2.apply\_access\_width\_criteria\_to\_non\_excl\_stores**

Apply the given exclusive store width matching criteria to non-exclusive stores.

Type: `bool`

Default value: `true`



**exclusive\_monitor0\_2.clear\_on\_strex\_address\_mismatch**

Whether monitor is cleared when strex fails due to address mismatch.

Type: `bool`

Default value: `true`

**exclusive\_monitor0\_2.enable\_component**

Enable component.

Type: `bool`

Default value: `true`

**exclusive\_monitor0\_2.exclusive\_monitor\_clear\_on\_atomic\_from\_same\_master**

Monitor atomics from the same master.

Type: `bool`

Default value: `true`

**exclusive\_monitor0\_2.log2\_granule\_size**

log2 of address granule size.

Type: `uint32_t`

Default value: 3

**exclusive\_monitor0\_2.match\_access\_width**

Fail STREX if not the same access width as LDREX.

Type: `bool`

Default value: `true`

**exclusive\_monitor0\_2.match\_secure\_state**

Treat the secure state like an address bit.

Type: `bool`

Default value: `true`

**exclusive\_monitor0\_2.monitor\_access\_level**

0: Monitor all accesses, 1: Monitor all accesses except WriteBack, 2: Only monitor accesses with memory type NonCacheable or Device.

Type: `uint32_t`

Default value: 0

**exclusive\_monitor0\_2.monitor\_non\_excl\_stores**

Monitor non-exclusive stores from the same master.

Type: bool

Default value: false

**exclusive\_monitor0\_2.number\_of\_monitors**

Number of monitors.

Type: unsigned

Default value: 8

**exclusive\_monitor0\_2.shareability\_domain**

Maximum shareability domain of interest, transactions outside of the domain will pass through un-monitored (0-non-shared, 1-inner, 2-outer, 3-system).

Type: unsigned

Default value: 3

**exclusive\_monitor0\_3.apply\_access\_width\_criteria\_to\_non\_excl\_stores**

Apply the given exclusive store width matching criteria to non-exclusive stores.

Type: bool

Default value: true

**exclusive\_monitor0\_3.clear\_on\_strex\_address\_mismatch**

Whether monitor is cleared when strex fails due to address mismatch.

Type: bool

Default value: true

**exclusive\_monitor0\_3.enable\_component**

Enable component.

Type: bool

Default value: true

**exclusive\_monitor0\_3.exclusive\_monitor\_clear\_on\_atomic\_from\_same\_master**

Monitor atomics from the same master.

Type: `bool`

Default value: `true`

### **`exclusive_monitor0_3.log2_granule_size`**

log2 of address granule size.

Type: `uint32_t`

Default value: 3

### **`exclusive_monitor0_3.match_access_width`**

Fail STREX if not the same access width as LDREX.

Type: `bool`

Default value: `true`

### **`exclusive_monitor0_3.match_secure_state`**

Treat the secure state like an address bit.

Type: `bool`

Default value: `true`

### **`exclusive_monitor0_3.monitor_access_level`**

0: Monitor all accesses, 1: Monitor all accesses except WriteBack, 2: Only monitor accesses with memory type NonCacheable or Device.

Type: `uint32_t`

Default value: 0

### **`exclusive_monitor0_3.monitor_non_excl_stores`**

Monitor non-exclusive stores from the same master.

Type: `bool`

Default value: `false`

### **`exclusive_monitor0_3.number_of_monitors`**

Number of monitors.

Type: `unsigned`

Default value: 8

**exclusive\_monitor0\_3.shareability\_domain**

Maximum shareability domain of interest, transactions outside of the domain will pass through un-monitored (0-non-shared, 1-inner, 2-outer, 3-system).

Type: unsigned

Default value: 3

**exclusive\_monitor1\_0.apply\_access\_width\_criteria\_to\_non\_excl\_stores**

Apply the given exclusive store width matching criteria to non-exclusive stores.

Type: bool

Default value: true

**exclusive\_monitor1\_0.clear\_on\_strex\_address\_mismatch**

Whether monitor is cleared when strex fails due to address mismatch.

Type: bool

Default value: true

**exclusive\_monitor1\_0.enable\_component**

Enable component.

Type: bool

Default value: true

**exclusive\_monitor1\_0.exclusive\_monitor\_clear\_on\_atomic\_from\_same\_master**

Monitor atomics from the same master.

Type: bool

Default value: true

**exclusive\_monitor1\_0.log2\_granule\_size**

log2 of address granule size.

Type: uint32\_t

Default value: 3

**exclusive\_monitor1\_0.match\_access\_width**

Fail STREX if not the same access width as LDREX.

Type: bool

Default value: `true`

**exclusive\_monitor1\_0.match\_secure\_state**

Treat the secure state like an address bit.

Type: `bool`

Default value: `true`

**exclusive\_monitor1\_0.monitor\_access\_level**

0: Monitor all accesses, 1: Monitor all accesses except WriteBack, 2: Only monitor accesses with memory type NonCacheable or Device.

Type: `uint32_t`

Default value: `0`

**exclusive\_monitor1\_0.monitor\_non\_excl\_stores**

Monitor non-exclusive stores from the same master.

Type: `bool`

Default value: `false`

**exclusive\_monitor1\_0.number\_of\_monitors**

Number of monitors.

Type: `unsigned`

Default value: `8`

**exclusive\_monitor1\_0.shareability\_domain**

Maximum shareability domain of interest, transactions outside of the domain will pass through un-monitored (0-non-shared, 1-inner, 2-outer, 3-system).

Type: `unsigned`

Default value: `3`

**exclusive\_monitor1\_1.apply\_access\_width\_criteria\_to\_non\_excl\_stores**

Apply the given exclusive store width matching criteria to non-exclusive stores.

Type: `bool`

Default value: `true`

**exclusive\_monitor1\_1.clear\_on\_strex\_address\_mismatch**

Whether monitor is cleared when strex fails due to address mismatch.

Type: `bool`

Default value: `true`

**exclusive\_monitor1\_1.enable\_component**

Enable component.

Type: `bool`

Default value: `true`

**exclusive\_monitor1\_1.exclusive\_monitor\_clear\_on\_atomic\_from\_same\_master**

Monitor atomics from the same master.

Type: `bool`

Default value: `true`

**exclusive\_monitor1\_1.log2\_granule\_size**

log2 of address granule size.

Type: `uint32_t`

Default value: 3

**exclusive\_monitor1\_1.match\_access\_width**

Fail STREX if not the same access width as LDREX.

Type: `bool`

Default value: `true`

**exclusive\_monitor1\_1.match\_secure\_state**

Treat the secure state like an address bit.

Type: `bool`

Default value: `true`

**exclusive\_monitor1\_1.monitor\_access\_level**

0: Monitor all accesses, 1: Monitor all accesses except WriteBack, 2: Only monitor accesses with memory type NonCacheable or Device.

Type: `uint32_t`

Default value: 0

**exclusive\_monitor1\_1.monitor\_non\_excl\_stores**

Monitor non-exclusive stores from the same master.

Type: bool

Default value: false

**exclusive\_monitor1\_1.number\_of\_monitors**

Number of monitors.

Type: unsigned

Default value: 8

**exclusive\_monitor1\_1.shareability\_domain**

Maximum shareability domain of interest, transactions outside of the domain will pass through un-monitored (0-non-shared, 1-inner, 2-outer, 3-system).

Type: unsigned

Default value: 3

**exclusive\_monitor1\_2.apply\_access\_width\_criteria\_to\_non\_excl\_stores**

Apply the given exclusive store width matching criteria to non-exclusive stores.

Type: bool

Default value: true

**exclusive\_monitor1\_2.clear\_on\_strex\_address\_mismatch**

Whether monitor is cleared when strex fails due to address mismatch.

Type: bool

Default value: true

**exclusive\_monitor1\_2.enable\_component**

Enable component.

Type: bool

Default value: true

**exclusive\_monitor1\_2.exclusive\_monitor\_clear\_on\_atomic\_from\_same\_master**

Monitor atomics from the same master.

Type: `bool`

Default value: `true`

### **`exclusive_monitor1_2.log2_granule_size`**

log2 of address granule size.

Type: `uint32_t`

Default value: 3

### **`exclusive_monitor1_2.match_access_width`**

Fail STREX if not the same access width as LDREX.

Type: `bool`

Default value: `true`

### **`exclusive_monitor1_2.match_secure_state`**

Treat the secure state like an address bit.

Type: `bool`

Default value: `true`

### **`exclusive_monitor1_2.monitor_access_level`**

0: Monitor all accesses, 1: Monitor all accesses except WriteBack, 2: Only monitor accesses with memory type NonCacheable or Device.

Type: `uint32_t`

Default value: 0

### **`exclusive_monitor1_2.monitor_non_excl_stores`**

Monitor non-exclusive stores from the same master.

Type: `bool`

Default value: `false`

### **`exclusive_monitor1_2.number_of_monitors`**

Number of monitors.

Type: `unsigned`

Default value: 8



**exclusive\_monitor1\_2.shareability\_domain**

Maximum shareability domain of interest, transactions outside of the domain will pass through un-monitored (0-non-shared, 1-inner, 2-outer, 3-system).

Type: unsigned

Default value: 3

**exclusive\_monitor1\_3.apply\_access\_width\_criteria\_to\_non\_excl\_stores**

Apply the given exclusive store width matching criteria to non-exclusive stores.

Type: bool

Default value: true

**exclusive\_monitor1\_3.clear\_on\_strex\_address\_mismatch**

Whether monitor is cleared when strex fails due to address mismatch.

Type: bool

Default value: true

**exclusive\_monitor1\_3.enable\_component**

Enable component.

Type: bool

Default value: true

**exclusive\_monitor1\_3.exclusive\_monitor\_clear\_on\_atomic\_from\_same\_master**

Monitor atomics from the same master.

Type: bool

Default value: true

**exclusive\_monitor1\_3.log2\_granule\_size**

log2 of address granule size.

Type: uint32\_t

Default value: 3

**exclusive\_monitor1\_3.match\_access\_width**

Fail STREX if not the same access width as LDREX.

Type: bool

Default value: `true`

**exclusive\_monitor1\_3.match\_secure\_state**

Treat the secure state like an address bit.

Type: `bool`

Default value: `true`

**exclusive\_monitor1\_3.monitor\_access\_level**

0: Monitor all accesses, 1: Monitor all accesses except WriteBack, 2: Only monitor accesses with memory type NonCacheable or Device.

Type: `uint32_t`

Default value: `0`

**exclusive\_monitor1\_3.monitor\_non\_excl\_stores**

Monitor non-exclusive stores from the same master.

Type: `bool`

Default value: `false`

**exclusive\_monitor1\_3.number\_of\_monitors**

Number of monitors.

Type: `unsigned`

Default value: `8`

**exclusive\_monitor1\_3.shareability\_domain**

Maximum shareability domain of interest, transactions outside of the domain will pass through un-monitored (0-non-shared, 1-inner, 2-outer, 3-system).

Type: `unsigned`

Default value: `3`

**revision**

Revision.

Type: `string`

Default value: `"r1p2"`

## 3.297 PL350\_SMC\_NAND\_FLASH

Defined in `LISA/PL350_SMC_NAND_FLASH.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r1p2	Full support
r2p2	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### About PL350\_SMC\_NAND\_FLASH

Program the component as you would the hardware.

### Iris and MTI instances for PL350\_SMC\_NAND\_FLASH

This model has the following Iris instances:

Name	Instance type
PL350_SMC_NAND_FLASH	PL350_SMC_NAND_FLASH
PL350_SMC_NAND_FLASH.busslave	PVBusSlave

This model has the following MTI trace components:

Name	Component type
PL350_SMC_NAND_FLASH.busslave	PVBusSlave

### Ports for PL350\_SMC\_NAND\_FLASH

Port	Direction	Protocol	Description
irq	master	Signal	Interrupt signaling.
pvbus	slave	PVBus	Slave port for connection to PV bus master/decoder.

### Parameters for PL350\_SMC\_NAND\_FLASH

#### DEVICE\_1

Device manufacturer code.

Type: `uint32_t`

Default value: `0xEC`

#### DEVICE\_2

Device code.

Type: `uint32_t`

Default value: 0xDA

**DEVICE\_3**

Device 3rd cycle code.

Type: uint32\_t

Default value: 0x80

**DEVICE\_4**

Device 4th cycle code.

Type: uint32\_t

Default value: 0x15

**DEVICE\_NAME**

Device Name.

Type: string

Default value: "Samsung K9F1G08U0M"

**NAND\_BLOCK\_COUNT**

number of blocks in the flash device.

Type: uint32\_t

Default value: 2048

**NAND\_FLASH\_SIZE**

flash size in byte.

Type: uint32\_t

Default value: 0x10800000

**NAND\_PAGE\_COUNT\_PER\_BLOCK**

number of pages in each block.

Type: uint32\_t

Default value: 64

**NAND\_PAGE\_SIZE**

page size.

Type: uint32\_t

Default value: 2112

#### **NAND\_SPARE\_SIZE\_PER\_PAGE**

Spare size per page.

Type: uint32\_t

Default value: 64

#### **NAND\_VALID\_SIZE\_PER\_PAGE**

valid page size.

Type: uint32\_t

Default value: 2048

## 3.298 PL370\_HDLCD

Defined in `LISA/PL370_HDLCD.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [Quality level definitions](#).



Too fast a pixel clock can slow the rest of the simulation.

### Iris and MTI instances for PL370\_HDLCD

This model has the following Iris instances:

Name	Instance type
PL370_HDLCD	PL370_HDLCD
PL370_HDLCD.busmaster	PVBusMaster
PL370_HDLCD.busslave	PVBusSlave
PL370_HDLCD.timer	ClockTimerThread
PL370_HDLCD.timer.timer	ClockTimerThread64
PL370_HDLCD.timer.timer.thread	SchedulerThread
PL370_HDLCD.timer.timer.thread_event	SchedulerThreadEvent

This model has the following MTI trace components:

Name	Component type
PL370_HDLCDC.busmaster	PVBusMaster
PL370_HDLCDC.busslave	PVBusSlave

### Ports for PL370\_HDLCDC

Port	Direction	Protocol	Description
clk_in	slave	ClockSignal	Master clock input, typically 24MHz, to drive pixel clock timing.
display_trace	master	FrameTracingProtocol	Test/Debug Frame Capture connection.
display	master	LCD	Connection to visualization component.
intr	master	Signal	Interrupt signaling line for flyback events.
pvb主_m	master	PVBus	DMA port for collecting video data from memory/framebuffer.
pvb主	slave	PVBus	Slave port for connection to PV bus master/decoder.

### Parameters for PL370\_HDLCDC

#### diagnostics

Diagnostics level.

Type: uint32\_t

Default value: 0

#### disable\_snooping\_dma

Disable DMA snooping.

Type: bool

Default value: false

#### force\_frame\_rate

Force frame rate to the value of the parameter in frames per simulated second, regardless of the input clock. When 0, use the input clock as a pixel clock.

Type: int

Default value: 50

## 3.299 PL390\_GIC

Defined in LISA/PL390\_GIC.lisa.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

## About PL390\_GIC

The GIC provides support for three interrupt types:

- Software Generated Interrupts (SGI)
- Private Peripheral Interrupts (PPI)
- Shared Peripheral Interrupts (SPI)

You can set:

- Security state for an interrupt
- Priority state for an interrupt
- Enabling or disabling state for an interrupt
- Processors that receive an interrupt

A processor interface consists of a pair of interfaces called `pvbuss_cpu` and `pvbuss_distributor`. The `enable_cx` and `match_cx` signals identify the originator of a transaction on `pvbuss_cpu`. Similarly, the `enable_dx` and `match_dx` signals identify the originator of a transaction on `pvbuss_distributor`. `X` corresponds to the number of a processor interface.



Note

To reduce compile time, the registers are not available by default. To activate them, uncomment either of the following statements in `PL390_GIC.lisa`:

```
// #define FEW_CADI_REGISTER
// #define ALL_CADI_REGISTER
```

## Iris and MTI instances for PL390\_GIC

This model has the following Iris instances:

Name	Instance type
PL390_GIC	PL390_GIC
PL390_GIC.busslave_cpu	PVBusSlave
PL390_GIC.busslave_distributor	PVBusSlave

This model has the following MTI trace components:

Name	Component type
PL390_GIC.busslave_cpu	PVBusSlave
PL390_GIC.busslave_distributor	PVBusSlave

## Ports for PL390\_GIC

Port	Direction	Protocol	Description
cfgsdisable	slave	Signal	Set preventing write accesses to security-critical configuration registers.
enable_c	slave	ValueState	Compared with masked PVBUS manager id to select processor interface: (manager_id & enable_c<n>) == match_c<n>.
enable_d	slave	ValueState	Compared with masked PVBUS manager id to select distributor interface: (manager_id & enable_d<n>) == match_d<n>.
legacy_nfiq	slave	Signal	Legacy FIQ interrupt for processor Interface <n>.
legacy_nirq	slave	Signal	Input interrupt signals.
match_c	slave	ValueState	Mask on the PVBUS manager id to select processor interface: (manager_id & enable_c<n>) == match_c<n>.
match_d	slave	ValueState	Mask on the PVBUS manager id to select distributor interface: (manager_id & enable_d<n>) == match_d<n>.
nfiq	master	Signal	Send out FIQ signal to processor <n>.
nirq	master	Signal	Send out IRQ signal to processor <n>.
ppi_c0	slave	Signal	Private peripheral interrupt for processor 0 (num_cpus> = 1).
ppi_c1	slave	Signal	Private peripheral interrupt for processor 1 (num_cpus> = 2).
ppi_c2	slave	Signal	Private peripheral interrupt for processor 2 (num_cpus> = 3).
ppi_c3	slave	Signal	Private peripheral interrupt for processor 3 (num_cpus> = 4).
ppi_c4	slave	Signal	Private peripheral interrupt for processor 4 (num_cpus> = 5).
ppi_c5	slave	Signal	Private peripheral interrupt for processor 5 (num_cpus> = 6).
ppi_c6	slave	Signal	Private peripheral interrupt for processor 6 (num_cpus> = 7).
ppi_c7	slave	Signal	Private peripheral interrupt for processor 7 (num_cpus> = 8).
pvbuss_cpu	slave	PVBUS	Slave port for connection to processor interface.
pvbuss_distributor	slave	PVBUS	Slave port for connection to distributor interface.
reset_in	slave	Signal	Reset signal.
spi	slave	Signal	Shared peripheral interrupt inputs.

## Parameters for PL390\_GIC

### ARCHITECTURE\_VERSION

set architecture version in periph\_id register.

Type: `int`

Default value: 1

### AXI\_IF

set interface type in peripheral identification register 8.

Type: `bool`

Default value: `true`



**C\_ID\_WIDTH**

width of the cpu interface manager id.

Type: `int`

Default value: `32`

**D\_ID\_WIDTH**

width of the distributor interface manager id.

Type: `int`

Default value: `32`

**ENABLE\_LEGACY\_FIQ**

provide legacy fiq interrupt inputs.

Type: `bool`

Default value: `true`

**ENABLE\_LEGACY\_IRQ**

provide legacy irq interrupt inputs.

Type: `bool`

Default value: `true`

**ENABLE\_PPI\_EDGE**

ppi edge sensitive.

Type: `bool`

Default value: `false`

**ENABLE\_TRUSTZONE**

support trustzone.

Type: `bool`

Default value: `true`

**INIT\_ENABLE\_C0**

initial value of register ENABLE\_C0.

Type: `int`

Default value: `0xffffffff`

**INIT\_ENABLE\_C1**

initial value of register ENABLE\_C1.

Type: int

Default value: 0xffffffff

**INIT\_ENABLE\_C2**

initial value of register ENABLE\_C2.

Type: int

Default value: 0xffffffff

**INIT\_ENABLE\_C3**

initial value of register ENABLE\_C3.

Type: int

Default value: 0xffffffff

**INIT\_ENABLE\_C4**

initial value of register ENABLE\_C4.

Type: int

Default value: 0xffffffff

**INIT\_ENABLE\_C5**

initial value of register ENABLE\_C5.

Type: int

Default value: 0xffffffff

**INIT\_ENABLE\_C6**

initial value of register ENABLE\_C6.

Type: int

Default value: 0xffffffff

**INIT\_ENABLE\_C7**

initial value of register ENABLE\_C7.

Type: int

Default value: 0xffffffff

**INIT\_ENABLE\_D0**

initial value of register ENABLE\_D0.

Type: `int`

Default value: `0xffffffff`

**INIT\_ENABLE\_D1**

initial value of register ENABLE\_D1.

Type: `int`

Default value: `0xffffffff`

**INIT\_ENABLE\_D2**

initial value of register ENABLE\_D2.

Type: `int`

Default value: `0xffffffff`

**INIT\_ENABLE\_D3**

initial value of register ENABLE\_D3.

Type: `int`

Default value: `0xffffffff`

**INIT\_ENABLE\_D4**

initial value of register ENABLE\_D4.

Type: `int`

Default value: `0xffffffff`

**INIT\_ENABLE\_D5**

initial value of register ENABLE\_D5.

Type: `int`

Default value: `0xffffffff`

**INIT\_ENABLE\_D6**

initial value of register ENABLE\_D6.

Type: `int`

Default value: `0xffffffff`

**INIT\_ENABLE\_D7**

initial value of register ENABLE\_D7.

Type: `int`

Default value: `0xffffffff`

**INIT\_MATCH\_C0**

initial value of register MATCH\_C0.

Type: `int`

Default value: `0`

**INIT\_MATCH\_C1**

initial value of register MATCH\_C1.

Type: `int`

Default value: `1`

**INIT\_MATCH\_C2**

initial value of register MATCH\_C2.

Type: `int`

Default value: `2`

**INIT\_MATCH\_C3**

initial value of register MATCH\_C3.

Type: `int`

Default value: `3`

**INIT\_MATCH\_C4**

initial value of register MATCH\_C4.

Type: `int`

Default value: `4`

**INIT\_MATCH\_C5**

initial value of register MATCH\_C5.

Type: `int`

Default value: `5`

**INIT\_MATCH\_C6**

initial value of register MATCH\_C6.

Type: `int`

Default value: 6

**INIT\_MATCH\_C7**

initial value of register MATCH\_C7.

Type: `int`

Default value: 7

**INIT\_MATCH\_D0**

initial value of register MATCH\_D0.

Type: `int`

Default value: 0

**INIT\_MATCH\_D1**

initial value of register MATCH\_D1.

Type: `int`

Default value: 1

**INIT\_MATCH\_D2**

initial value of register MATCH\_D2.

Type: `int`

Default value: 2

**INIT\_MATCH\_D3**

initial value of register MATCH\_D3.

Type: `int`

Default value: 3

**INIT\_MATCH\_D4**

initial value of register MATCH\_D4.

Type: `int`

Default value: 4

**INIT\_MATCH\_D5**

initial value of register MATCH\_D5.

Type: `int`

Default value: 5

**INIT\_MATCH\_D6**

initial value of register MATCH\_D6.

Type: `int`

Default value: 6

**INIT\_MATCH\_D7**

initial value of register MATCH\_D7.

Type: `int`

Default value: 7

**NUM\_CPU**

number of cpu interfaces.

Type: `int`

Default value: 8

**NUM\_LSPI**

number of lockable shared peripheral interrupts.

Type: `int`

Default value: 31

**NUM\_PPI**

number of peripheral interrupts.

Type: `int`

Default value: 16

**NUM\_PRIORITY\_LEVELS**

number of priority levels.

Type: `int`

Default value: 256

**NUM\_SGI**

number of software generated interrupts.

Type: int

Default value: 16

**NUM\_SPI**

number of shared peripheral interrupts.

Type: int

Default value: 988

## 3.300 PLLClockControl

Defined in LISA/PLLClockControl.lisa.

### About PLLClockControl

Clock Rate Control.

### Iris and MTI instances for PLLClockControl

This model has the following Iris instances:

Name	Instance type
PLLClockControl	PLLClockControl
PLLClockControl.clock_ctl	ClockDivider
PLLClockControl.pllclk_div	ClockDivider

This model has the following MTI trace components:

Name	Component type
PLLClockControl.clock_ctl	ClockDivider
PLLClockControl.pllclk_div	ClockDivider

### Ports for PLLClockControl

Port	Direction	Protocol	Description
clk_en	slave	Signal	-
clk_in	slave	ClockSignal	-
clk_out	master	ClockSignal	-
clk_rate	slave	ClockRateControl	-
clk_sel	slave	Value	-
dvfs_freq_in	slave	ValueState	-

Port	Direction	Protocol	Description
lock	master	Signal	-
refclk_in	slave	ClockSignal	-
unlock	master	Signal	-

## Parameters for PLLClockControl

### diagnostics

Diagnostics.

Type: uint32\_t

Default value: 0

## 3.301 PLLControl

Defined in LISA/PLLControl.lisa.

### About PLLControl

Simulate PLL clock frequency control logic.

### Iris and MTI instances for PLLControl

This model has the following Iris instances:

Name	Instance type
PLLControl	PLLControl
PLLControl.clkdiv	ClockDivider

This model has the following MTI trace components:

Name	Component type
PLLControl.clkdiv	ClockDivider

## Ports for PLLControl

Port	Direction	Protocol	Description
clk_in	slave	ClockSignal	-
clk_out	master	ClockSignal	-
lock	master	Signal	-
rate	slave	ClockRateControl	-
unlock	master	Signal	-

## Parameters for PLLControl

This component does not have any parameters.



## 3.302 PMU

Defined in `LISA/PMU.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
0	Alpha support

For an explanation of the quality levels, see [Quality level definitions](#).

### About PMU

PMU (Performance Monitoring Unit).

### Iris and MTI instances for PMU

This model has the following Iris instances:

Name	Instance type
PMU	<a href="#">PMU</a>

### Ports for PMU

Port	Direction	Protocol	Description
<code>apb_bus_s</code>	slave	PVBus	-
<code>clk_in</code>	slave	<a href="#">ClockSignal</a>	-

### Parameters for PMU

#### **diagnostics**

Diagnostics.

Type: `uint8_t`

Default value: 2

#### **feat\_rme**

RME Support.

Type: `bool`

Default value: `false`

#### **is\_amu**

AMU instance.

Type: `bool`

Default value: `false`

### **num\_monitors**

Number of PMU monitors.

Type: `uint8_t`

Default value: 1

### **pm\_64bit\_ext**

64bit programmer view extension.

Type: `bool`

Default value: `false`

### **pm\_dual\_page\_ext**

Dual page in APB address space.

Type: `bool`

Default value: `false`

### **pm\_edgedetect\_ext**

Edge detect.

Type: `bool`

Default value: `false`

### **pm\_export\_ext**

Event output - exported event.

Type: `bool`

Default value: `false`

### **pm\_fzo\_ext**

Freeze on overflow.

Type: `bool`

Default value: `false`

### **pm\_mpam\_filter\_ext**

MPAM filtering.

Type: `bool`

Default value: `false`

### **`pm_oac_ext`**

Observability and access control.

Type: `bool`

Default value: `false`

### **`pm_sos_filter_ext`**

Secure operating state filtering.

Type: `bool`

Default value: `true`

### **`pm_sshot_ext`**

Snapshot.

Type: `bool`

Default value: `true`

### **`pm_threshold_ext`**

Threshold.

Type: `bool`

Default value: `false`

### **`pm_tro_ext`**

Trace Interface.

Type: `bool`

Default value: `false`

### **`pmevfiltr2_present`**

Event filtering registers 2 present.

Type: `bool`

Default value: `false`

### **`pmevfiltr_present`**

Event filtering registers present.

Type: `bool`

Default value: `false`

**pmimpdef\_present**

Implementation defined register present.

Type: `bool`

Default value: `false`

**pmoflow\_present**

Overflow interrupt present.

Type: `bool`

Default value: `false`

### 3.303 PPUMTWakerequest

Defined in `LISA/PPUMTWakerequest.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

**About PPUMTWakerequest**

Power Policy Unit (PPU) v8.2 Multi-threaded Core Wakerequest Logic.

**Ports for PPUMTWakerequest**

Port	Direction	Protocol	Description
cpu_pchannel_m	master	PChannel	-
ppu_pchannel_s	slave	PChannel	-
thread_wake_request	slave	Signal	-
wakerequest	master	Signal	-

**Parameters for PPUMTWakerequest**

**mt\_mode**

Multi-threaded mode.

Type: `bool`

Default value: `false`

**thread0\_op\_mode\_bit**

Thread0 Operation Mode bit of DEVPACTIVE.

Type: `uint32_t`

Default value: 16

**thread1\_op\_mode\_bit**

Thread1 Operation Mode bit of DEVPACTIVE.

Type: `uint32_t`

Default value: 17

3.304 PPUMultiThreadModeSwitch

Defined in `LISA/PPUMTWakerequest.lisa`.

About PPUMultiThreadModeSwitch

PPU mode switch between single-thread mode and multi-thread mode. Support up to 8 cores and thread number per core is no more than 2.

Ports for PPUMultiThreadModeSwitch

Port	Direction	Protocol	Description
pchannel_from_ppu_s	slave	PChannel	-
pchannel_to_cpu_m	master	PChannel	-
wakerequest_from_gic_s	slave	Signal	-
wakerequest_to_ppu_m	master	Signal	-

Parameters for PPUMultiThreadModeSwitch

**mt\_mode**

Multi-threaded mode.

Type: `bool`

Default value: `false`

## 3.305 PPUv0

Defined in `LISA/PPUv0.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### About PPUv0

Power Policy Unit (PPU) v0.8 architectural model.

### Iris and MTI instances for PPUv0

This model has the following Iris instances:

Name	Instance type
PPUv0	PPUv0
PPUv0.busslave	PVBusSlave

This model has the following MTI trace components:

Name	Component type
PPUv0	PPUv0
PPUv0.busslave	PVBusSlave

### Ports for PPUv0

Port	Direction	Protocol	Description
irq	master	Signal	-
powerdown	master	Signal	-
ppuhwstat	master	Value	-
pvbus_s	slave	PVBus	-
smpen	slave	Signal	-
standbywfi	slave	Signal	-
wakerequest	slave	Signal	-

### Parameters for PPUv0

#### **default\_power\_state\_on**

Default power state ON.

Type: `bool`

Default value: `false`

**device\_channels**

Number of device channels (0: P-Channel, 1-8: Q-Channels).

Type: `uint32_t`

Default value: 0

**dynamic\_off**

Dynamic Off.

Type: `bool`

Default value: `false`

**dynamic\_on**

Dynamic On.

Type: `bool`

Default value: `false`

**dynamic\_warm\_reset**

Dynamic Warm Reset.

Type: `bool`

Default value: `false`

**full\_ret**

Full Retention (0: not supported, 1: static, 2: dynamic).

Type: `uint32_t`

Default value: 0

**func\_ret**

Functional Retention (0: not supported, 1: static, 2: dynamic).

Type: `uint32_t`

Default value: 0

**logic\_ret**

Logic Retention (0: not supported, 1: static, 2: dynamic).

Type: `uint32_t`

Default value: 0

**mem\_off**

Memory Off (0: not supported, 1: static, 2: dynamic).

Type: `uint32_t`

Default value: 0

**mem\_ret**

Memory Retention (0: not supported, 1: static, 2: dynamic).

Type: `uint32_t`

Default value: 0

**revision**

Revision.

Type: `string`

Default value: "r0p0"

**use\_active\_signal**

Use device-active signal.

Type: `bool`

Default value: `false`

## 3.306 PPUv1

Defined in `LISA/PPUv1.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following ports were added:

- `dev_warm_clk_en_out`



## About PPUv1

Software can determine which features the PPU supports by reading the PPU Identification Register 0, `PPU_IDR0` and the PPU Identification Register 1, `PPU_IDR1`.

The following power policies are offered by the PPU model, in order of increasing priority:

- Off.
- Emulated Off.
- Memory Retention.
- Emulated Memory Retention.
- Logic Retention.
- Full Retention.
- Memory Off.
- Functional Retention.
- On.
- Warm Reset.
- Debug Recovery Reset.

For the power mode transition rules, see [Arm Power Policy Unit Architecture Specification](#).

There are 16 operating mode values. The meaning of these values is specific to the device that is connected to the PPU. The operating mode can only be configured to change during a power transition of ON to ON.

The PPU model supports static and dynamic transitions on the P-Channel interface. It does not yet support Q-Channel.

`DEVPACTIVE` and `DEVSTATE` have the following bit encodings:

### **DEVPACTIVE bits [10:0]**

Each bit indicates a required power mode.

### **DEVSTATE bits [3:0]**

The integer formed by this bitfield indicates a power mode.

### **DEVPACTIVE bits [23:16]**

Operating mode. The interpretation of these bits depends on the `DEVPACTIVE` use model (Ladder or Independent).

### **DEVSTATE bits [7:4]**

The integer formed by this bitfield indicates an operating mode.

Communication over the Low Power Interface (`PREQUEST` and `PACTIVE`) uses blocking calls and does not model any delays. See [PChannel protocol](#) for further details.

For the AMBA Low Power Interface Specification Arm Q-Channel and P-Channel Interfaces, see [AMBA Low Power Interface Specification](#).

For static transitions, software sets the policy as the required power mode. The PPU then sends a `REQUEST` with the required power state to the attached device. The device can `ACCEPT` or `DENY` it. For dynamic transitions, software sets the policy as a minimum power mode. Based on whether the device has sent a signal using `DEVPACTIVE`, the PPU sends a `PREQUEST` with the required power state to the attached device. The device can `ACCEPT` or `DENY` it.

The PPU model is automatically reset by the simulation engine when the model starts up. Reset can also occur through the `reset_in` port. The PPU model is reset only when the signal value is `signal::Set`. Use `signal::Set` instead of zero, its integer value, to prevent unexpected behavior.

The `ppuhwstat` port notifies the power state change inside the PPU and the definition of each bit is the same as `DEVPACTIVE[10:0]`.

The `smpen` and `standbywfi` ports are defined in PPUv0 and are not supported in PPUv1.

### Differences between the model and the RTL

- Q-Channel is not supported
- The PPU model has been validated with devices supporting only ON and OFF power modes. Arm has not tested the case where a connected device supports other power modes offered by the PPU.

### Iris and MTI instances for PPUv1

This model has the following Iris instances:

Name	Instance type
PPUv1	PPUv1
PPUv1.busslave	PVBusSlave

This model has the following MTI trace components:

Name	Component type
PPUv1	PPUv1
PPUv1.busslave	PVBusSlave

### Ports for PPUv1

Port	Direction	Protocol	Description
<code>dev_clk_en_out</code>	master	Signal	Domain clock enable
<code>dev_emu_clk_en_out</code>	master	Signal	Domain emulated mode clock enable
<code>dev_emu_isolaten_out</code>	master	Signal	Domain emulated isolation control.
<code>dev_isolaten_out</code>	master	Signal	Domain isolation control.
<code>dev_poresetn_out</code>	master	Signal	Domain power on reset
<code>dev_ret_reseten_out</code>	master	Signal	Domain retention reset.

Port	Direction	Protocol	Description
dev_warm_clk_en_out	master	Signal	Domain warm clock enable - applicable only when RST_CG_CFG configuration is non-zero
dev_warm_resetrn_out	master	Signal	Domain warm reset
devpactive	master	PChannel	P-Channel port
irq	master	Signal	PPU IRQ signal
powerdown	master	Signal	Notify whether or not the PPU is in OFF state.
ppuhwstat	master	Value	Notify the power state change inside the PPU. The definition of each bit is the same as DEVPACTIVE[10:0].
pdbus_s	slave	PVBus	PPU APB bus slave port
reset_in	slave	Signal	PPU reset signal input
wakerequest	slave	Signal	Input port for the wakerequest signal. It is ORed with PACTIVE[8] (ON) inside the PPU as input to PPU DEVPACTIVE[8] (ON). The "is_core_ppu" parameter controls whether there is additional logic to hold this signal until the PPU is in OFF/OFF_EMU state.

## Parameters for PPUv1

### RevD\_support

Whether to support RevD Locked IRQ.

Type: `bool`

Default value: `true`

### bypass\_handshake

Bypass pscm handshake.

Type: `bool`

Default value: `false`

### dbg\_recov

Debug Recovery Reset (0: not supported, 1: static mode only, 2: both dynamic & static mode).

Type: `uint32_t`

Default value: `0`

### default\_op\_dyn\_en

Whether to enable operating mode dynamic transition by default.

Type: `bool`

Default value: `false`

### default\_op\_policy

Default operating policy.

Type: `uint32_t`

Default value: `0`

### **default\_power\_state\_on**

Default power state ON.

Type: `bool`

Default value: `false`

### **default\_pwr\_dyn\_en**

Whether to enable dynamic power mode transition by default.

Type: `bool`

Default value: `false`

### **device\_channels**

Number of device channels (0: P-Channel, 1-8: Q-Channels).

Type: `uint32_t`

Default value: `0`

### **dynamic\_off**

Dynamic Off.

Type: `bool`

Default value: `false`

### **dynamic\_on**

Dynamic On.

Type: `bool`

Default value: `false`

### **dynamic\_warm\_reset**

Dynamic Warm Reset.

Type: `bool`

Default value: `false`

### **full\_ret**

Full Retention (0: not supported, 1: static mode only, 2: both dynamic & static mode).

Type: `uint32_t`

Default value: 0

### **func\_ret**

Functional Retention (0: not supported, 1: static only, 2: both dynamic & static mode).

Type: `uint32_t`

Default value: 0

### **is\_core\_ppu**

Set PPU to be Core\_PPU type (Core\_PPU has additional logic to hold wake\_request until it's in OFF/OFF\_EMU state).

Type: `bool`

Default value: `false`

### **lock\_support**

Whether to support LOCK feature.

Type: `bool`

Default value: `true`

### **logic\_ret**

Logic Retention (0: not supported, 1: static mode only, 2: both dynamic & static mode).

Type: `uint32_t`

Default value: 0

### **mem\_off**

Memory Off (0: not supported, 1: static mode only, 2: both dynamic & static mode).

Type: `uint32_t`

Default value: 0

### **mem\_ret**

Memory Retention (0: not supported, 1: static mode only, 2: both dynamic & static mode).

Type: `uint32_t`

Default value: 0

**mem\_ret\_emu**

Emulated Memory Retention (0: not supported, 1: static mode only, 2: both dynamic & static mode).

Type: `uint32_t`

Default value: 0

**num\_opmode\_cfg**

Number of operating modes.

Type: `uint32_t`

Default value: 0

**off\_emu**

Emulated Off (0: not supported, 1: static mode only, 2: both dynamic & static mode).

Type: `uint32_t`

Default value: 0

**off\_mem\_ret\_trans\_cfg**

OFF to MEM\_RET direct transition configuration (0: not allowed, 1: allowed).

Type: `bool`

Default value: `false`

**op\_active\_cfg**

Operating mode active configuration (0: Ladder use model, 1: Independent use model).

Type: `uint32_t`

Default value: 0

**revision**

Revision.

Type: `string`

Default value: `"r1p1"`

**use\_active\_signal**

Use device-active signal.

Type: `bool`

Default value: `false`

## 3.307 PPUv1\_Cluster\_Wakerequest\_Logic

Defined in `LISA/PPUv1_Cluster_Wakerequest_Logic.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### About PPUv1\_Cluster\_Wakerequest\_Logic

PPUv1 wake request stall logic.

### Ports for PPUv1\_Cluster\_Wakerequest\_Logic

Port	Direction	Protocol	Description
<code>cluster_wake_request</code>	master	Signal	-
<code>core_wake_request_in</code>	slave	Signal	-
<code>core_wake_request_out</code>	master	Signal	-
<code>ppuhwstat</code>	slave	Value	-
<code>reset_in</code>	slave	Signal	-

### Parameters for PPUv1\_Cluster\_Wakerequest\_Logic

#### **`core_ppu_wakerequest_stall_condition_after_reset`**

Set Stall Condition of Core WakeRequest (from GIC) for Core PPU after reset.

Type: `bool`

Default value: `false`

#### **`disable_core_ppu_wakerequest_input_stall`**

Disable wakerequest input stall of Core PPU. This feature is enabled by default to mimic the P-Channel request stall when Cluster PPU is in OFF.

Type: `bool`

Default value: `false`

#### **`enable_cluster_wakeup_if_cluster_on_funcret`**

enable cluster wakeup logic. If it's disabled, `core_wake_request_in[x]` will be directly connected to `core_wake_request_out[x]` and `cluster_wake_request` port is disabled.

Type: `bool`

Default value: `true`

## 3.308 PS2Keyboard

Defined in `LISA/PS2Keyboard.lisa`.

### About PS2Keyboard

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050\_KMI component.

### Iris and MTI instances for PS2Keyboard

This model has the following Iris instances:

Name	Instance type
PS2Keyboard	PS2Keyboard
PS2Keyboard.ps2_clocktimer	ClockTimerThread
PS2Keyboard.ps2_clocktimer.timer	ClockTimerThread64
PS2Keyboard.ps2_clocktimer.timer.thread	SchedulerThread
PS2Keyboard.ps2_clocktimer.timer.thread_event	SchedulerThreadEvent

### Ports for PS2Keyboard

Port	Direction	Protocol	Description
clk_in	slave	ClockSignal	-
keyboard	slave	KeyboardStatus	-
ps2	master	PS2Data	-

### Parameters for PS2Keyboard

This component does not have any parameters.

## 3.309 PS2Mouse

Defined in `LISA/PS2Mouse.lisa`.

### About PS2Mouse

Interface component, which takes the keypress/release signals from the Visualisation component and translates them into clocked PS2Data signals which can be routed to a PL050\_KMI component.



## Iris and MTI instances for PS2Mouse

This model has the following Iris instances:

Name	Instance type
PS2Mouse	PS2Mouse
PS2Mouse.ps2_clocktimer	ClockTimerThread
PS2Mouse.ps2_clocktimer.timer	ClockTimerThread64
PS2Mouse.ps2_clocktimer.timer.thread	SchedulerThread
PS2Mouse.ps2_clocktimer.timer.thread_event	SchedulerThreadEvent

## Ports for PS2Mouse

Port	Direction	Protocol	Description
clk_in	slave	ClockSignal	-
mouse	slave	MouseStatus	-
ps2	master	PS2Data	-

## Parameters for PS2Mouse

This component does not have any parameters.

## 3.310 PVBUS2AMBAPV

Defined in `examples/SystemCEExport/Bridges/PVBus2AMBAPV.lisa`.

### Changes in 11.31.15

The following parameters were added:

- `mpam-mec-attribute-transform`
- `transport-write-snoop`

## About PVBUS2AMBAPV



Variants of PVBUS2AMBAPV also exist with multiple input and output ports.

The AMBAPV protocol definition in LISA, `AMBAPVProtocol.lisa`, specifies a 64-bit bus width, so the PVBUS2AMBAPV bridge also handles a 64-bit bus width.

If you need to connect to a component that uses a bus interface with a smaller or larger bus width, the recommended method is to insert a downsizer or upsizer respectively.

Alternatively, you could define a new bus protocol with the required bit width, for example AMBAPV32, and update the corresponding bridges to use the new protocol on AMBA-PV ports:

```
master port<AMBAPV32> amba_pv_m
```

## Limitations

- The debug channel does not support all MTE operations, it only supports tag-only requests (for reading or writing tags only).
- For tag-only requests, the transaction may carry a null data pointer.

## Dumping the DMI cache

DMI viewer provides the debugging functionality of the PVBUS2AMBAPV bridge. When activated, it dumps the content of the DMI cache in the bridge in the following CSV format:

```
Range_start, Range_end_incl, Pointer, Latency, R/W, Attributes
```

To activate this functionality, a name for the counters output file must be set, using the `counters-file-name` parameter. If the counters file name is set, when `dump-dmi-cache` is set to 1 at runtime, the DMI cache of the bridge is dumped. The runtime parameter is always reset to 0 when the dump has completed.

## Iris and MTI instances for PVBUS2AMBAPV

This model has the following Iris instances:

Name	Instance type
PVBUS2AMBAPV	PVBUS2AMBAPV
PVBUS2AMBAPV.bus_bridge	PVBUSBridge

This model has the following MTI trace components:

Name	Component type
PVBUS2AMBAPV	PVBUS2AMBAPV
PVBUS2AMBAPV.bus_bridge	PVBUSBridge

## Ports for PVBUS2AMBAPV

Port	Direction	Protocol	Description
amba_pv_m	master	AMBAPV	-
pvbus_s	slave	PVBUS	-

## Parameters for PVBUS2AMBAPV

### counters-file-name

Prefix of the file name to store counters at the end of simulation.

Type: `string`

Default value: `""`

### **dump-dmi-cache**

Dumps the content of the DMI cache into a file.

Type: `bool`

Default value: `false`

### **dump-dmi-file-name**

Prefix of the file name to dump the content of the DMI when requested.

Type: `string`

Default value: `""`

### **force-dmi-size**

Force DMI start and end address to be 4kB-aligned.

Type: `bool`

Default value: `true`

### **min-range-to-cache**

Min DMI range size to cache in the bridge.

Type: `uint32_t`

Default value: `0x10000`

### **mpam-mec-attribute-transform**

User-defined transform to be applied to bus attributes like ManagerID, ExtendedID or UserFlags, for MPAM Attributes and/or MEC id encoded into bus attributes. For example, 'ExtendedID[62:55]=MPAM\_PMG, ExtendedID[54:39]=MPAM\_PARTID, ExtendedID[38]=MPAM\_SP[0], ExtendedID[37]=MPAM\_SP[1], UserFlags[31:16]=MECID'. An empty string disables MPAM and MEC id support.

Type: `string`

Default value: `""`

### **size**

Maximum size of memory region.

Type: `uint64_t`

Default value: `0x10000000000000`

### **transport-write-snoop**

Transport write-snoop request downstream.

Type: `bool`

Default value: `false`

## 3.311 PVBUS2AMBAPVACE

Defined in `examples/SystemCEExport/Bridges/PVBus2AMBAPVACE.lisa`.

### Changes in 11.31.15

The following parameters were added:

- `mpam-mec-attribute-transform`
- `transport-write-snoop`

### About PVBUS2AMBAPVACE

PVBus2AMBAPVACE depends on the AMBA-PV API, which must be at least version 1.4.

The translation of bus transactions by the bridge has some impact on performance. Bus masters that cache memory transactions avoid much of this impact.

### Limitations

- The debug channel does not support all MTE operations, it only supports tag-only requests (for reading or writing tags only).
- For tag-only requests, the transaction may carry a null data pointer.

### Dumping the DMI cache

DMI viewer provides the debugging functionality of the PVBus2AMBAPV bridge. When activated, it dumps the content of the DMI cache in the bridge in the following CSV format:

```
Range start, Range end incl, Pointer, Latency, R/W, Attributes
```

To activate this functionality, set a name for the counters output file using the `counters-file-name` parameter. If the counters file name is set, when `dump-dmi-cache` is set to 1 at runtime, the DMI cache of the bridge is dumped. The runtime parameter is always reset to 0 when the dump has completed.

### Iris and MTI instances for PVBUS2AMBAPVACE

This model has the following Iris instances:

Name	Instance type
PVBus2AMBAPVACE	PVBus2AMBAPVACE
PVBus2AMBAPVACE.bus_bridge	PVBusBridge
PVBus2AMBAPVACE.pvbus_tlm_switch	PVBusMapper

This model has the following MTI trace components:

Name	Component type
PVBus2AMBAPVACE	PVBus2AMBAPVACE
PVBus2AMBAPVACE.bus_bridge	PVBusBridge
PVBus2AMBAPVACE.pvbus_tlm_switch	PVBusMapper

### Ports for PVBus2AMBAPVACE

Port	Direction	Protocol	Description
amba_pv_ace_m	master	AMBAPVACE	-
pvbus_over_tlm_control	slave	PVBusOverTLMControl	-
pvbus_s	slave	PVBus	-

### Parameters for PVBus2AMBAPVACE

#### **counters-file-name**

Prefix of the file name to store counters at the end of simulation.

Type: `string`

Default value: `""`

#### **dmi-cache-name**

DEPRECATED: This parameter will be ignored. Name of the DMI cache. Useful for multiple bridges to share the same cache.

Type: `string`

Default value: `""`

#### **dump-dmi-cache**

Dumps the content of the DMI cache into a file.

Type: `bool`

Default value: `false`

#### **dump-dmi-file-name**

Prefix of the file name to dump the content of the DMI when requested.

Type: `string`

Default value: ""

### **force-dmi-size**

Force DMI start and end address to be 4kB-aligned.

Type: `bool`

Default value: `true`

### **min-range-to-cache**

Min DMI range size to cache in the bridge.

Type: `uint32_t`

Default value: `0x10000`

### **mpam-mec-attribute-transform**

User-defined transform to be applied to bus attributes like ManagerID, ExtendedID or UserFlags, for MPAM Attributes and/or MEC id encoded into bus attributes. For example, 'ExtendedID[62:55]=MPAM\_PMG, ExtendedID[54:39]=MPAM\_PARTID, ExtendedID[38]=MPAM\_SP[0], ExtendedID[37]=MPAM\_SP[1], UserFlags[31:16]=MECID'. An empty string disables MPAM and MEC id support.

Type: `string`

Default value: ""

### **route-tlm**

Route all the PVBUS traffic explicitly to the TLM bus. Allows to monitor transactions on the TLM bus but slows down the emulation. The routing must always be to TLM if there is not a corresponding AMBAPVACE2PVBUS bridge downstream.

Type: `bool`

Default value: `true`

### **route-tlm-filter**

Route TLM filter set a range (or multiple ranges) of addresses that will use PVBUS even if `route-tlm` is set to `true`.

The `route-tlm-filter` is specified in JSON format. Example,

```
[
  {
    "begin": 0x2f000000,
    "size": 0x1000
  },
  {
    "begin": 0x4f000000,
    "size": 0x2000
  }
]
```

```
    }  
  ]  
}
```

Type: `string`

Default value: `""`

**set-ace-lite**

Set bridge mode when connecting to ace-lite ports. If true, the bridge will not deal with SNOOPS.

Type: `bool`

Default value: `false`

**size**

Maximum size of memory region, i.e. the first unsupported address.

Type: `uint64_t`

Default value: `0x100000000000000`

**transport-write-snoop**

Transport write-snoop request downstream.

Type: `bool`

Default value: `true`

### 3.312 PVBUS2AMBAPVx4

Defined in `examples/SystemCEExport/Bridges/PVBus2AMBAPVx4.lisa`.

**About PVBUS2AMBAPVx4**

PVBus to AMBA-PV protocol converter with array size 4.

**Iris and MTI instances for PVBUS2AMBAPVx4**

This model has the following Iris instances:

Name	Instance type
PVBus2AMBAPVx4	PVBus2AMBAPVx4
PVBus2AMBAPVx4.pvbus2ambapv_U (where U = 0–3)	PVBus2AMBAPV
PVBus2AMBAPVx4.pvbus2ambapv_U.bus_bridge (where U = 0–3)	PVBusBridge

This model has the following MTI trace components:

Name	Component type
PVBus2AMBAPVx4.pvbus2ambapv_U (where $U = 0-3$ )	PVBus2AMBAPV
PVBus2AMBAPVx4.pvbus2ambapv_U.bus_bridge (where $U = 0-3$ )	PVBusBridge

### Ports for PVBus2AMBAPVx4

Port	Direction	Protocol	Description
amba_pv_m	master	AMBAPV	-
pvbus_s	slave	PVBus	-

### Parameters for PVBus2AMBAPVx4

This component does not have any parameters.

## 3.313 PVBus2AMBAPVx8

Defined in `examples/SystemCEexport/Bridges/PVBus2AMBAPVx8.lisa`.

### About PVBus2AMBAPVx8

PVBus to AMBA-PV protocol converter with array size 8.

### Iris and MTI instances for PVBus2AMBAPVx8

This model has the following Iris instances:

Name	Instance type
PVBus2AMBAPVx8	PVBus2AMBAPVx8
PVBus2AMBAPVx8.pvbus2ambapv_U (where $U = 0-7$ )	PVBus2AMBAPV
PVBus2AMBAPVx8.pvbus2ambapv_U.bus_bridge (where $U = 0-7$ )	PVBusBridge

This model has the following MTI trace components:

Name	Component type
PVBus2AMBAPVx8.pvbus2ambapv_U (where $U = 0-7$ )	PVBus2AMBAPV
PVBus2AMBAPVx8.pvbus2ambapv_U.bus_bridge (where $U = 0-7$ )	PVBusBridge

### Ports for PVBus2AMBAPVx8

Port	Direction	Protocol	Description
amba_pv_m	master	AMBAPV	-
pvbus_s	slave	PVBus	-

### Parameters for PVBus2AMBAPVx8

This component does not have any parameters.



## 3.314 PVBUS4KBTo1KBSplitter

Defined in `LISA/PVBUS4KBTo1KBSplitter.lisa`.

### About PVBUS4KBTo1KBSplitter

The purpose of this component is to allow an upstream component to access four downstream components in the same 4 KB address range. It splits the 4 KB range from 0 to 0xfff into the following four 1 KB ranges, which allows four different components to be attached to the 4 KB range:

- 0x0-0x3ff
- 0x400-0x7ff
- 0x800-0xbff
- 0xc00-0xfff

This overcomes a limitation of PVBUS which only allows components to be attached to memory addresses that are a multiple of 4 KB in size.



The forwarded transactions have their address re-aligned with the 1 KB boundary in the range 0x0-0x3ff. For example, address 0x0402 becomes address 0x002 of the second peripheral, which is the one attached to `pvb_m[1]`.

### Limitations

Unaligned transactions that cross the boundaries between two peripherals are not supported. For example, when unaligned transactions are enabled by your models, you can access two double words at address 0x03ed, but you cannot access two double words at address 0x3f7.

### Iris and MTI instances for PVBUS4KBTo1KBSplitter

This model has the following Iris instances:

Name	Instance type
PVBUS4KBTo1KBSplitter	PVBUS4KBTo1KBSplitter
PVBUS4KBTo1KBSplitter.input_slave	PVBUSSlave
PVBUS4KBTo1KBSplitter.output_masterZ (where Z = 0-3)	PVBUSMaster

This model has the following MTI trace components:

Name	Component type
PVBUS4KBTo1KBSplitter.input_slave	PVBUSSlave
PVBUS4KBTo1KBSplitter.output_masterZ (where Z = 0-3)	PVBUSMaster

## Ports for PVBus4KBTo1KBSplitter

Port	Direction	Protocol	Description
pvbus_m	master	PVBus	The four downstream ports to be connected to peripherals. Each port covers 1KiB of the address space. Output address on each port will be in the range 0x0 - 0x03FF.
pvbus_s	slave	PVBus	The upstream port. Accepts addresses in range 0x0 - 0x0FFF. Outside of this range transactions will abort.

## Parameters for PVBus4KBTo1KBSplitter

This component does not have any parameters.

## 3.315 PVBusBridge

Defined in `LISA/PVBusBridge.lisa`.

### Changes in 11.31.15

The following parameters were added:

- `disable-write-snooper`

### About PVBusBridge

A PVBusBridge bridges incoming transactions to a PVDevice port.

### Iris and MTI instances for PVBusBridge

This model has the following Iris instances:

Name	Instance type
PVBusBridge	PVBusBridge

This model has the following MTI trace components:

Name	Component type
PVBusBridge	PVBusBridge

## Ports for PVBusBridge

Port	Direction	Protocol	Description
control	slave	PVBusBridgeControl	Control signal.
device	master	PVDevice	Optimised connection out to devices.
dump_dmi	slave	Signal	On the assert of this signal the bridge will dump dmi cache content into a csv file
pvbus_s	slave	PVBus	Connection in from bus master.
reset	slave	Signal	On the assert of this signal, a reset of the bus slave will be latched this is used by the bus deadlock detection logic.

## Parameters for PVBusBridge

### **counters-file-name**

Prefix of the file name to store counters at the end of simulation.

Type: `string`

Default value: `""`

### **disable-write-snooper**

Disable the write-snooper in this bridge.

Type: `bool`

Default value: `false`

### **dmi-cache-name**

DEPRECATED: This parameter will be ignored. Name of the DMI cache in the bridge. Useful for multiple bridges to share the same cache.

Type: `string`

Default value: `""`

### **dump-dmi-file-name**

Prefix of the file name to dump the content of the DMI when requested.

Type: `string`

Default value: `""`

### **min-range-to-cache**

Min DMI range size to cache in the bridge.

Type: `uint32_t`

Default value: `0x10000`

## 3.316 PVBusCache

Defined in `LISA/PVBusCache.lisa`.

### About PVBusCache

This component defines parameters and ports that are private, subject to change, and should not be used outside of the PL310 model.

## Ports for PVBusCache

Port	Direction	Protocol	Description
bus_in	slave	PVBus	Connections in from bus master.
bus_out	master	PVBus	Connections out to bus slaves.
control	slave	PVBusCacheControl	Configuration and control port.
device	master	PVBusCacheDevice	Connection out to cache device.

## Parameters for PVBusCache

### line\_count

Number of cache lines to manage.

Type: uint32\_t

Default value: N/A

### line\_size

Size of cache lines in bytes.

Type: uint32\_t

Default value: 32

## 3.317 PVBusDecoder

Defined in `LISA/PVBusDecoder.lisa`.

### About PVBusDecoder

Each slave connection is associated with a specific address range on the `pvbus_m_range` port. In LISA+, the syntax for this is:

```
decoder.pvbus_m_range[start..end] = slave.pvbus
```

The values for start (inclusive) and end (inclusive) must specify a 4KB-aligned region of a multiple of 4K bytes. You can specify an address range for the slave, where the decoder remaps addresses into the appropriate range. The default address range for a slave is `[0-(sizeofMasterRange - 1)]`.

Examples of usage:

```
component PlatformDecoder
{
    slave port<PVBus> pvbus_s;
    master port<PVBus> sdram;
    master port<PVBus> flash;
    master port<PVBus> uart;

    composition
```

```

{
    pvdecoder : PVBusDecoder;
}

connection
{
    self.pvbus_s => pvdecoder.pvbus_s;
    pvdecoder.pvbus_m_range[0x000000..0x0fffff] => sdram;
    pvdecoder.pvbus_m_range[0x100000..0x1fffff] => flash;
    pvdecoder.pvbus_m_range[0x200000..0x2fffff] => uart;
    pvdecoder.pvbus_m_range[0xff0000..0xffffffff] => sdram[0x070000..0x07ffff];
}
}

```

### Ports for PVBusDecoder

Port	Direction	Protocol	Description
pvbus_m_range	master	PVBus	Specifies the address range for the bus master. The range must be 4KB aligned and a multiple of 4KB in size. If the address range is larger than the size of the slave device, the slave is aliased.
pvbus_s	slave	PVBus	Accepts incoming transactions. Connect this port to a bus master, or to the output of another bus decoder.

### Parameters for PVBusDecoder

This component does not have any parameters.

## 3.318 PVBusExclusiveMonitor

Defined in `LISA/PVBusExclusiveMonitor.lisa`.

### About PVBusExclusiveMonitor

Global exclusive monitor.

### Iris and MTI instances for PVBusExclusiveMonitor

This model has the following Iris instances:

Name	Instance type
PVBusExclusiveMonitor	<a href="#">PVBusExclusiveMonitor</a>
PVBusExclusiveMonitor.bus_mapper	<a href="#">PVBusMapper</a>

This model has the following MTI trace components:

Name	Component type
PVBusExclusiveMonitor	<a href="#">PVBusExclusiveMonitor</a>
PVBusExclusiveMonitor.bus_mapper	<a href="#">PVBusMapper</a>

### Ports for PVBusExclusiveMonitor

Port	Direction	Protocol	Description
excl_cleared	master	<a href="#">Signal</a>	Exclusive monitor clear signal port.

Port	Direction	Protocol	Description
pvbus_m	master	PVBus	Bus master port.
pvbus_s	slave	PVBus	Bus slave port.

## Parameters for PVBusExclusiveMonitor

### **apply\_access\_width\_criteria\_to\_non\_excl\_stores**

Apply the given exclusive store width matching criteria to non-exclusive stores.

Type: `bool`

Default value: `true`

### **clear\_on\_strex\_address\_mismatch**

Whether monitor is cleared when strex fails due to address mismatch.

Type: `bool`

Default value: `true`

### **enable\_component**

Enable component.

Type: `bool`

Default value: `true`

### **exclusive\_monitor\_clear\_on\_atomic\_from\_same\_master**

Monitor atomics from the same master.

Type: `bool`

Default value: `true`

### **log2\_granule\_size**

log2 of address granule size.

Type: `uint32_t`

Default value: `0`

### **match\_access\_width**

Fail STREX if not the same access width as LDREX.

Type: `bool`

Default value: `false`

**match\_secure\_state**

Treat the secure state like an address bit.

Type: `bool`

Default value: `true`

**monitor\_access\_level**

Which accesses to monitor:

**0**

Monitor all accesses

**1**

Monitor all accesses except WriteBack

**2**

Only monitor accesses with memory type NonCacheable or Device.

Type: `uint32_t`

Default value: `0`

**monitor\_non\_excl\_stores**

Monitor non-exclusive stores from the same master.

Type: `bool`

Default value: `false`

**number\_of\_monitors**

Number of monitors.

Type: `unsigned`

Default value: `8`

**shareability\_domain**

Maximum shareability domain of interest, transactions outside of the domain will pass through un-monitored

**0**

non-shared

**1**

inner

**2**

outer

**3**

system.

Type: unsigned

Default value: 3

## 3.319 PVBusExclusiveSquasher

Defined in `LISA/PVBusExclusiveSquasher.lisa`.

### About PVBusExclusiveSquasher

PVBusExclusiveSquasher modifies any exclusive transactions that pass through it so that they will be treated as regular transactions by any components downstream on the bus.

This is intended to be used to model M-class cores that have EXREQx and EXRESPx signals configured as follows:

- EXREQx is unconnected
- EXRESPx is tied low

This effectively forces all exclusive transactions to succeed. This is a legitimate thing to do if only the local exclusive monitor inside the core is required to monitor exclusive transactions. In other words, the system contains no other masters with access to the RAM.

### Iris and MTI instances for PVBusExclusiveSquasher

This model has the following Iris instances:

Name	Instance type
PVBusExclusiveSquasher	PVBusExclusiveSquasher
PVBusExclusiveSquasher.bus_modifier	PVBusMapper

This model has the following MTI trace components:

Name	Component type
PVBusExclusiveSquasher.bus_modifier	PVBusMapper

### Ports for PVBusExclusiveSquasher

Port	Direction	Protocol	Description
pdbus_m	master	PVBus	-
pdbus_s	slave	PVBus	-

### Parameters for PVBusExclusiveSquasher

This component does not have any parameters.



## 3.320 PVBUSGICv3Comms

Defined in `LISA/GICv3CommsPVBUS.lisa`.

### About PVBUSGICv3Comms

GICv3 Component for conversion between GICv3Comms protocol and PVBUS.

### Iris and MTI instances for PVBUSGICv3Comms

This model has the following Iris instances:

Name	Instance type
PVBUSGICv3Comms	PVBUSGICv3Comms
PVBUSGICv3Comms.ExportTest.PVBUSGICv3Comms.pvbus_m[0].pvbusmaster	PVBUSMaster
PVBUSGICv3Comms.bus_slave	PVBUSSlave

This model has the following MTI trace components:

Name	Component type
PVBUSGICv3Comms	PVBUSGICv3Comms
PVBUSGICv3Comms.ExportTest.PVBUSGICv3Comms.pvbus_m[0].pvbusmaster	PVBUSMaster
PVBUSGICv3Comms.bus_slave	PVBUSSlave

### Ports for PVBUSGICv3Comms

Port	Direction	Protocol	Description
axi_manager_id_m	master	Value_64	-
distributor_m	master	GICv3Comms	-
pvbus_m	master	PVBUS	-
pvbus_s	slave	PVBUS	-

### Parameters for PVBUSGICv3Comms

This component does not have any parameters.

## 3.321 PVBUSLogger

Defined in `LISA/PVBUSLogger.lisa`.

### About PVBUSLogger

A PVBUSLogger has a slave and a master port and traffic is passed straight through. All traffic is logged using an MTI trace event.

## Iris and MTI instances for PVBUSLogger

This model has the following Iris instances:

Name	Instance type
PVBUSLogger	PVBUSLogger
PVBUSLogger.mapper	PVBUSMapper

This model has the following MTI trace components:

Name	Component type
PVBUSLogger	PVBUSLogger
PVBUSLogger.mapper	PVBUSMapper

## Ports for PVBUSLogger

Port	Direction	Protocol	Description
pvbuss_m	master	PVBUS	Bus master port.
pvbuss_s	slave	PVBUS	Bus slave port.

## Parameters for PVBUSLogger

### **trace\_debug**

Enable tracing of debug transactions.

Type: `bool`

Default value: `false`

### **trace\_snoops**

Enable tracing of ACE snoop requests.

Type: `bool`

Default value: `false`

## 3.322 PVBUSMapper

Defined in `LISA/PVBUSMapper.lisa`.

### About PVBUSMapper

This component is similar to `PVBUSModifier`, but in addition:

- It has multiple downstream ports
- It allows routing of transactions to any one of these ports
- It allows arbitrary remapping of transaction addresses and attributes

As a generic modeling component, it does not have a hardware revision code.

For an example of how to use PVBusMapper, SEE `$PVLIB_HOME/examples/LISAPlus/RemappingWithPVBusMapper/`.

## Iris and MTI instances for PVBusMapper

This model has the following Iris instances:

Name	Instance type
PVBusMapper	PVBusMapper

This model has the following MTI trace components:

Name	Component type
PVBusMapper	PVBusMapper

## Ports for PVBusMapper

Port	Direction	Protocol	Description
control	master	PVBusMapperControl	Configuration port to determine mappings.
pdbus_m	master	PVBus	Bus master ports.
pdbus_s	slave	PVBus	Bus slave port.
reset	slave	Signal	Reset signal.

## Parameters for PVBusMapper

### handling\_of\_dvm\_messages\_from\_downstream

What to do with DVM (Distributed Virtual Memory) messages received from downstream. The options are to 'forward' them upstream unaltered, to 'terminate' them, or to 'handle' them locally and get called through `handleDownstreamDVMMessage()`.

Type: string

Default value: "forward"

### handling\_of\_dvm\_messages\_from\_upstream

What to do with DVM (Distributed Virtual Memory) messages received from upstream. The options are to 'forward' them downstream unaltered, to 'terminate' them, or to 'handle' them locally and get called through `handleUpstreamDVMMessage()`.

Type: string

Default value: "forward"

### handling\_of\_upstream\_busmapchanged

What to do with BusMapChanged events from downstream. The options are to 'forward' them upstream or to 'use\_remapdecisiongroup' to propagate the event upstream. For almost all cases the

default should not be changed. The latter is required only for specific topologies with loop in multi-interconnect systems.

Type: `string`

Default value: `"forward"`

### **handling\_of\_upstream\_snoop\_requests**

What to do with snoop requests from downstream. The options are to 'forward', 'terminate' or 'handle'. NOTE that currently the snoop request addresses are *not* translated and so if your device alters the address translation then you will almost certainly want to 'terminate'.

Type: `string`

Default value: `"forward"`

## **3.323 PVBusMaster**

Defined in `LISA/PVBusMaster.lisa`.

### **About PVBusMaster**

The `PVBusMaster` subcomponent allows a device to generate PVBus transactions. It does this by providing a control port that allows a component to instantiate `pv::TransactionGenerator` objects. These objects can be used to generate bus transactions.

See `PVTransactionMasterProtocol.lisa` for details.

A bus mastering component should connect the `pvbus_m` port to its own bus port.

Example:

```
component DmaTransfer
{
  master port<PVBus> pvbus_m;
  master port<PVTransactionMaster> busmaster_control;
  composition {
    busmaster : PVBusMaster;
  }
  resources {
    pv::TransactionGenerator* stream_in;
    pv::TransactionGenerator* stream_out;
  }
  connection {
    busmaster.pvbus_m => self.pvbus_m;
    self.busmaster_control => busmaster.control;
  }
  behaviour init() {
    stream_in = busmaster_control.createTransactionGenerator();
    stream_out = busmaster_control.createTransactionGenerator();
    composition.init();
  }
  behaviour terminate() {
    delete stream_in;
    delete stream_out;
  }
}
```

```
composition.terminate();
}
behaviour transfer(pv::bus_addr_t start,
                  pv::bus_addr_t end,
                  pv::bus_addr_t destination)
{
    uint32_t data;
    bool ok = true;
    while (ok && start < end) {
        ok = stream_in->read32(start, &data);
        if (ok) {
            ok = stream_out->write32(destination, &data);
        }
        start += 4;
        destination += 4;
    }
}
```

Iris and MTI instances for PVBusMaster

This model has the following Iris instances:

Name	Instance type
PVBusMaster	PVBusMaster

This model has the following MTI trace components:

Name	Component type
PVBusMaster	PVBusMaster

Ports for PVBusMaster

Port	Direction	Protocol	Description
control	slave	PVTransactionMaster	Enables the owning component to instantiate pv::TransactionGenerator objects.
pvbus_m	master	PVBus	Sends out generated transactions to the bus.
reset	slave	Signal	On the de-assert of this signal, a reset of the bus master will be latched this is used by the bus deadlock detection logic.

Parameters for PVBusMaster

This component does not have any parameters.

3.324 PVBusModifier

Defined in LISA/PVBusModifier.lisa.

About PVBusModifier

Allow the connections to be modified through the component.

When a transaction is made to a 4 KiB address region, then the transaction is made through a channel, and if one doesn't exist then it must create one. The channel creation request is made

with the specific attributes of the transaction and it is up to the system to determine where the end point of that channel should be.

This component allows you to intercept the channel creation process and change the attributes for that channel as it flows through this component.

For example, you could remap the address, or the attributes, or both.



Channels are created and destroyed for any reason and so for a simulation to be deterministic then the component should always remap channels idempotently.

### Iris and MTI instances for PVBUSModifier

This model has the following Iris instances:

Name	Instance type
PVBUSModifier	PVBUSMapper

This model has the following MTI trace components:

Name	Component type
PVBUSModifier	PVBUSMapper

### Ports for PVBUSModifier

Port	Direction	Protocol	Description
control	master	PVBUSMapperControl	Configuration port to determine mappings.
pvbuse_m	master	PVBUS	Bus master port.
pvbuse_s	slave	PVBUS	Bus slave port.
reset	slave	Signal	Reset signal.

### Parameters for PVBUSModifier

#### handling\_of\_dvm\_messages\_from\_downstream

What to do with DVM (Distributed Virtual Memory) messages received from downstream. The options are to 'forward' them upstream unaltered, to 'terminate' them, or to 'handle' them locally and get called through handleDownstreamDVMMMessage().

Type: string

Default value: "forward"

**handling\_of\_dvm\_messages\_from\_upstream**

What to do with DVM (Distributed Virtual Memory) messages received from upstream. The options are to 'forward' them downstream unaltered, to 'terminate' them, or to 'handle' them locally and get called through `handleUpstreamDVMMMessage()`.

Type: `string`  
  
Default value: `"forward"`

**handling\_of\_upstream\_busmapchanged**

What to do with `BusMapChanged` events from downstream. The options are to 'forward' them upstream or to 'use\_remapdecisiongroup' to propagate the event upstream. For almost all cases the default should not be changed. The latter is required only for specific topologies with loop in multi-interconnect systems.

Type: `string`  
  
Default value: `"forward"`

**handling\_of\_upstream\_snoop\_requests**

What to do with snoop requests from downstream. The options are to 'forward', 'terminate' or 'handle'. NOTE that currently the snoop request addresses are *not* translated and so if your device alters the address translation then you will almost certainly want to 'terminate'.

Type: `string`  
  
Default value: `"forward"`

3.325 PVBUSMODIFIERX2

Defined in `LISA/PVBUSMODIFIERX2.lisa`.

About PVBUSMODIFIERX2

`PVBUSMODIFIERX2` is identical to `PVBUSMODIFIER`, except it has two downstream ports.

Iris and MTI instances for PVBUSMODIFIERX2

This model has the following Iris instances:

Name	Instance type
<code>PVBUSMODIFIERX2</code>	<a href="#">PVBUSMAPPER</a>

This model has the following MTI trace components:

Name	Component type
<code>PVBUSMODIFIERX2</code>	<a href="#">PVBUSMAPPER</a>

## Ports for PVBusModifierx2

Port	Direction	Protocol	Description
control	master	PVBusMapperControl	Configuration port to determine mappings.
pdbus_m	master	PVBus	Bus master ports.
pdbus_s	slave	PVBus	Bus slave port.
reset	slave	Signal	Reset signal.

## Parameters for PVBusModifierx2

### handling\_of\_dvm\_messages\_from\_downstream

What to do with DVM (Distributed Virtual Memory) messages received from downstream. The options are to 'forward' them upstream unaltered, to 'terminate' them, or to 'handle' them locally and get called through handleDownstreamDVMMMessage().

Type: string

Default value: "forward"

### handling\_of\_dvm\_messages\_from\_upstream

What to do with DVM (Distributed Virtual Memory) messages received from upstream. The options are to 'forward' them downstream unaltered, to 'terminate' them, or to 'handle' them locally and get called through handleUpstreamDVMMMessage().

Type: string

Default value: "forward"

### handling\_of\_upstream\_busmapchanged

What to do with BusMapChanged events from downstream. The options are to 'forward' them upstream or to 'use\_remapdecisiongroup' to propagate the event upstream. For almost all cases the default should not be changed. The latter is required only for specific topologies with loop in multi-interconnect systems.

Type: string

Default value: "forward"

### handling\_of\_upstream\_snoop\_requests

What to do with snoop requests from downstream. The options are to 'forward', 'terminate' or 'handle'. NOTE that currently the snoop request addresses are *not* translated and so if your device alters the address translation then you will almost certainly want to 'terminate'.

Type: string

Default value: "forward"



## 3.326 PVBusRouter

Defined in `LISA/PVBusRouter.lisa`.

### About PVBusRouter

Allow transactions to be routed arbitrarily.

### Iris and MTI instances for PVBusRouter

This model has the following Iris instances:

Name	Instance type
<code>PVBusRouter</code>	<a href="#">PVBusRouter</a>
<code>PVBusRouter.mapper</code>	<a href="#">PVBusMapper</a>

This model has the following MTI trace components:

Name	Component type
<code>PVBusRouter.mapper</code>	<a href="#">PVBusMapper</a>

### Ports for PVBusRouter

Port	Direction	Protocol	Description
<code>control</code>	master	<a href="#">PVBusRouterControl</a>	Configuration port to determine filters.
<code>pvbus_m</code>	master	<a href="#">PVBus</a>	Bus master ports.
<code>pvbus_s</code>	slave	<a href="#">PVBus</a>	Bus slave port.

### Parameters for PVBusRouter

This component does not have any parameters.

## 3.327 PVBusSlave

Defined in `LISA/PVBusSlave.lisa`.

### About PVBusSlave

Any component that acts as a bus slave must:

- Provide a `PVBus` slave port.
- Instantiate a `PVBusSlave` subcomponent, with the size parameter configured for the address range covered by the device.
- Connect the slave port to the `pvbus_s` port on the `PVBusSlave`.

A `PVBusSlave` handles incoming transactions, and handles support for mapping regions of device address space to work as RAM/ROM/device memory.

See `PVBusSlaveControlProtocol.lisa` for details of the mechanisms for configuring the memory regions.

The `PVBusSlave size` parameter controls the addressable size of the device. Addresses outside of this range will wrap around.

By default, the entire device address range is treated as device memory, meaning that all accesses will be routed to the device port. A component implementing device registers should connect the device port to a slave port that implements the `read()` and `write()` behaviors. (See the first example below).

A component that wants to implement regions of RAM or ROM must use the control port to reconfigure the `PVBusSlave`'s decoding. See the second example below.

Example of usage:

```
component BitLatch
{
  resources
  {
    flag : bool;
  }
  slave port<PVBus> pvbus_s;

  slave port<PVDevice> device_port
  {
    behaviour read(pv::ReadTransaction tx)
    {
      if (tx.getAddress() != 0)
      {
        return tx.generateAbort();
      }
      return tx.write8(flag ? 1 : 0);
    }
    behaviour write(pv::WriteTransaction tx)
    {
      if (tx.getAddress() != 0)
      {
        return tx.generateAbort();
      }
      flag = ((tx.read8() & 1) != 0);
      return tx.writeComplete();
    }
  }

  behavior debugRead(pv::ReadTransaction tx) : pv::Tx_Result
  {
    return device_port.read(tx);
  }

  behavior debugWrite(pv::WriteTransaction tx) : pv::Tx_Result
  {
    return device_port.write(tx);
  }
}

composition
{
  busslave : PVBusSlave(size=0x1000);
}

connection
{
  self.pvbus_s => busslave.pvbus_s;
  busslave.device => self.device_port;
}
```

```

}

component RAM
{
    slave port<PVBus> pvbus_s;
    master port<PVBusSlaveControl> busslave_control;

    composition
    {
        busslave : PVBusSlave(size=0x01000000);
    }
    connection
    {
        self.pvbus_s => busslave.pvbus_s;
        self.busslave_control => busslave.control;
    }
    behavior init()
    {
        busslave_control.setAccess(0, 0x01000000, pv::ACCESSTYPE_RW,
        pv::ACCESSMODE_MEMORY);
    }
}

```

## Iris and MTI instances for PVBusSlave

This model has the following Iris instances:

Name	Instance type
PVBusSlave	PVBusSlave

This model has the following MTI trace components:

Name	Component type
PVBusSlave	PVBusSlave

## Ports for PVBusSlave

Port	Direction	Protocol	Description
control	slave	<a href="#">PVBusSlaveControl</a>	Enables the owning component to control which regions of the device memory are to be handled as RAM/ROM/Device. These settings can be changed dynamically. For example, when a Flash component is being programmed, it can switch to treating reads as Device requests instead of ROM requests.
device	master	<a href="#">PVDevice</a>	Passes on requests for peripheral register accesses to permit the owning component to handle the request.
pvbus_s	slave	<a href="#">PVBus</a>	Handles incoming requests from bus masters.
reset	slave	<a href="#">Signal</a>	On the assert of this signal, a reset of the bus slave will be latched this is used by the bus deadlock detection logic.

## Parameters for PVBusSlave

### max\_access\_width

Maximum width of an access in bytes. Must be a power of 2. Wider accesses will split in chunks no larger than this.

Type: uint32\_t

Default value: 8

**read\_latency**

Memory read latency (ps/byte).

Type: `uint64_t`

Default value: 0

**size**

Addressable range of device (0 means 2^64).

Type: `uint64_t`

Default value: 0

**write\_latency**

Memory write latency (ps/byte).

Type: `uint64_t`

Default value: 0

### 3.328 PVCoherentInterconnect

Defined in `LISA/PVCoherentInterconnect.lisa`.

**About PVCoherentInterconnect**

`PVCoherentInterconnect` is a component written in LISA+ that is designed to be a generic interconnect.

`PVCoherentInterconnect` supports up to 128 clusters and requires minimal configuration, greatly simplifying the effort needed to set up the interconnect for systems with a high cluster count. It can also be modified to extend the number of clusters that can be connected to 4096. It has a single downstream port to handle the bus traffic.

You can find how to use this component in the [Fast Models Tutorials](#)

**Iris and MTI instances for PVCoherentInterconnect**

This model has the following Iris instances:

Name	Instance type
<code>PVCoherentInterconnect</code>	<a href="#">PVCoherentInterconnect</a>
<code>PVCoherentInterconnect.pvcacheX</code> (where $X = 0-1$ )	<a href="#">PVCache64</a>
<code>PVCoherentInterconnect.pvcacheX.downstream[Y].pvbusmaster</code> (where $X = 0-1$ ; $Y = 0-15$ )	<a href="#">PVBusMaster</a>

Name	Instance type
PVCoherentInterconnect.pvcacheX.upstream[Y] (where X = 0–1; Y = 0–63)	PVBusSlave
PVCoherentInterconnect.pvcache_common	PVCache64
PVCoherentInterconnect.pvcache_common.downstream[X].pvbusmaster (where X = 0–15)	PVBusMaster
PVCoherentInterconnect.pvcache_common.upstream[X] (where X = 0–63)	PVBusSlave

This model has the following MTI trace components:

Name	Component type
PVCoherentInterconnect.pvcacheX (where X = 0–1)	PVCache64
PVCoherentInterconnect.pvcacheX.downstream[Y].pvbusmaster (where X = 0–1; Y = 0–15)	PVBusMaster
PVCoherentInterconnect.pvcacheX.upstream[Y] (where X = 0–1; Y = 0–63)	PVBusSlave
PVCoherentInterconnect.pvcache_common	PVCache64
PVCoherentInterconnect.pvcache_common.downstream[X].pvbusmaster (where X = 0–15)	PVBusMaster
PVCoherentInterconnect.pvcache_common.upstream[X] (where X = 0–63)	PVBusSlave

### Ports for PVCoherentInterconnect

Port	Direction	Protocol	Description
downstream	master	PVBus	-
upstream	slave	PVBus	-

### Parameters for PVCoherentInterconnect

#### **cache\_state\_modelled**

Model the cache state to enable coherency in the interconnect. All the upstream components should have their cache state modelling on, for this to be on.

Type: `bool`

Default value: `true`

## 3.329 PVMemoryProtectionEngine

Defined in `LISA/PVMemoryProtectionEngine.lisa`.

### About PVMemoryProtectionEngine

PVMemoryProtectionEngine is a simplified implementation of a Memory Protection Engine (MPE) component as described in [Arm Realm Management Extension \(RME\) System Architecture](#).

PVMemoryProtectionEngine supports the following features:

- Memory encryption.
- Each 4KiB page in memory is encrypted based on an encryption key. Each Physical Address Space (PAS) has a separate encryption key.

- Two or more encryption keys can be the same value.
- Configurable encryption keys for each PAS.
- Configurable encryption block size.
- Configurable corruption strategy. You can control the behavior of memory contents that are not written by the access within the encryption block.
- Encryption/decryption algorithm is a simple XOR of data with the corresponding encryption key.
- Downstream memory is always stored as plain text, allowing debuggers to view data.

For example, if a block is currently encrypted by the ns-PAS and then a byte is written by the rl-PAS, if the `block_size_in_bytes` is 4KiB, the rest of the data in the 4KiB page is corrupted such that even if you read a different byte back through the ns-PAS, you would not get the original data.

The primary use case for this component is to identify software mis-programming, where the same Physical address is accessed through more than one PAS. With `PVMemoryProtectionEngine` enabled, a PE sees encrypted or corrupted data when it is accessed using a different PAS to the original PAS that wrote to that page in memory.

The `PVMemoryProtectionEngine` component is expected to be connected in a platform at the Point of Physical Aliasing (PoPA) if storage is shared, otherwise before each specific storage for a subset of the PASes.

`PVMemoryProtectionEngine` imposes a runtime cost when enabled. Normally, it is only needed when debugging and verifying the Realm Management Monitor (RMM) software. If the RMM software is correct, memory contents encrypted with the wrong key would not be visible.

The `PVMemoryProtectionEngine` does not encrypt or corrupt the tag data for MTE, but this feature will be supported in future.

## Iris and MTI instances for `PVMemoryProtectionEngine`

This model has the following Iris instances:

Name	Instance type
<code>PVMemoryProtectionEngine</code>	<code>PVMemoryProtectionEngine</code>
<code>PVMemoryProtectionEngine.mapper</code>	<code>PVBusMapper</code>
<code>PVMemoryProtectionEngine.mpe_bus_manager[0].pvbusmaster</code>	<code>PVBusMaster</code>

This model has the following MTI trace components:

Name	Component type
<code>PVMemoryProtectionEngine</code>	<code>PVMemoryProtectionEngine</code>
<code>PVMemoryProtectionEngine.mapper</code>	<code>PVBusMapper</code>
<code>PVMemoryProtectionEngine.mpe_bus_manager[0].pvbusmaster</code>	<code>PVBusMaster</code>

## Ports for PVMemoryProtectionEngine

Port	Direction	Protocol	Description
pvbuss_m	master	PVBus	Manager ports of the MPE
pvbuss_s	slave	PVBus	Subordinate port of the MPE

## Parameters for PVMemoryProtectionEngine

### **block\_size\_in\_bytes**

Encryption block size in bytes, supported sizes are 1 or 4096.

Type: `uint64_t`

Default value: 4096

### **corruption\_strategy**

Corruption strategy:

**0**

fill with constants per-old-encryption-context

**1**

fill with constants per-new-encryption-context

**2**

random data.

Type: `uint8_t`

Default value: 0

### **enable**

Enabling Memory Protection Engine.

Type: `bool`

Default value: `false`

### **ignore\_mecid**

Ignore MECID during encryption key calculation.

Type: `bool`

Default value: `false`

### **non\_secure\_pas\_enc\_key**

Non-Secure PAS encryption key.

Type: `uint8_t`

Default value: 0x22

### **output\_attributes\_parameter\_of\_core**

Encoding of various attributes on the bus.

Type: string

Default value: "ExtendedID[62:55]=MPAM\_PMG, ExtendedID[54:39]=MPAM\_PARTID, ExtendedID[38]=MPAM\_NS"

### **realm\_pas\_enc\_key**

Realm PAS encryption key.

Type: uint8\_t

Default value: 0x88

### **root\_pas\_enc\_key**

Root PAS encryption key.

Type: uint8\_t

Default value: 0x44

### **secure\_pas\_enc\_key**

Secure PAS encryption key.

Type: uint8\_t

Default value: 0x11

## 3.330 PVMetaDataController

Defined in `LISA/PVMetaDataController.lisa`.

### About PVMetaDataController

This component represents an entity in a memory system that responds to requests for manipulating metadata during a bus transaction and/or as part of Armv8.5-A instruction execution.

This component is intended to be an `sg::Component` which can be instantiated and connected in a platform.

MetaDataController is a `pv::RemapTransactionIntermediary` as it needs to intercept bus transactions to apply metadata operations and set up DMI to metadata memory, that is, assign `MetaDataPayload_t.set_dmi()`.



By being a `pv::RemapTransactionIntermediary` and using DMIs for data and metadata, it has a very small impact on simulation speed.



Note

The Armv8.5-A specification mentions that Colour-Check, which is a certain kind of metadata operation, happens in the physical memory system and a Processing Element (PE) only cares about the result of such an operation.

However, doing this for all memory accesses drastically slows down the simulation. While the first memory access of a `MetaDataPage_t` comes to `MetaDataController`, the rest of the accesses for addresses in this page are made through `MetaDataDMI_t`. This essentially means that Colour-Check is done in this class only once for every `MetaDataPage_t`. On other occasions, the intention is that the checks are done by the holder of `MetaDataDMI_t`.

## Interoperability requirements for non-Fast Models MTE-capable components

Fast Models components expect tags to be stored and accessed using the type `pv::MetaData::MetaDataElement_t`. If a simulation platform includes any Fast Models component that can access tags and MTE is enabled, all components that access tags must use `MetaDataElement_t`, or a compatible type such as `char`, `unsigned char`, or `std::byte`. For tag storage components, the tags must be provided as an array of `MetaDataElement_t`.

## Iris and MTI instances for PVMetaDataController

This model has the following Iris instances:

Name	Instance type
<code>PVMetaDataController</code>	<code>PVMetadataController</code>

## Ports for PVMetaDataController

Port	Direction	Protocol	Description
<code>pvbus_m</code>	master	<code>PVBus</code>	-
<code>pvbus_s</code>	slave	<code>PVBus</code>	-

## Parameters for PVMetaDataController

### `init_value`

Initialize metadata memory with this value. If one of `init_values_json` or `init_values_json_file` is specified this value applies only to any metadata not specified in the JSON.

Type: `uint8_t`

Default value: `0xd`

### `init_values_json`

A JSON value describing initial metadata values. Mutually exclusive with `init_values_json_file`.

The format is as follows:

```
{ "regions": [{ "begin": 0x0,
                "end": 0x10000,
                "mte_tag": 0xa},
              { "begin": 0x20000,
                "end": 0x50000,
                "mte_tag": 0xc}]
}
```

Type: string

Default value: ""

### **init\_values\_json\_file**

Path to a JSON file with initial metadata values. Mutually exclusive with `init_values_json`. The format is as follows:

```
{ "regions": [{ "begin": 0x0,
                "end": 0x10000,
                "mte_tag": 0xa},
              { "begin": 0x20000,
                "end": 0x50000,
                "mte_tag": 0xc}]
}
```

Type: string

Default value: ""

### **is\_enabled**

If false, disables the MetaData controller functionality, and makes the component invisible to passing transactions.

Type: bool

Default value: false

### **mte\_tag\_carveout\_json**

JSON string that specifies the PA range of the tag carveout regions and the beginning of the PA range for which they provide tag storage.

If `pa_regions_with_metadata_storage` defines which regions can have metadata, the tag carveout regions cannot overlap them, and each tagged region must be entirely covered by one of them.

The block size must be  $\geq 64$  bytes and a power of 2, defaulting to 4KiB. The maximum block size supported is 4KiB.

The carveout region size must be  $\geq 4\text{KiB}$  and a power of 2, and determines the size of the corresponding tagged region.

```
{ "regions": [{ "begin": 0x0,      "tag_carveout_region": [0xffffffff00000,
0xffffffff00fff]},
{ "begin": 0x20000, "tag_carveout_region": [0xffffffff01000,
0xffffffff01fff], "block_size": 0x100},
{ "begin": 0x100000, "tag_carveout_region": [0xffffffff08000,
0xffffffff0Bfff], "block_size": 0x2000}]}
```

Type: string

Default value: ""

### **mte\_tag\_carveout\_json\_file**

Path to a file which contains the JSON string that specifies the PA range of the tag carveout regions and the beginning of the PA range for which they provide tag storage.

If `pa_regions_with_metadata_storage` defines which regions can have metadata, the tag carveout regions cannot overlap them, and each tagged region must be entirely covered by one of them.

The block size must be  $\geq 64$  bytes and a power of 2, defaulting to 4KiB. The maximum block size supported is 4KiB.

The carveout region size must be  $\geq 4\text{KiB}$  and a power of 2, and determines the size of the corresponding tagged region.

```
{ "regions": [{ "begin": 0x0,      "tag_carveout_region": [0xffffffff00000,
0xffffffff00fff]},
{ "begin": 0x20000, "tag_carveout_region": [0xffffffff01000,
0xffffffff01fff], "block_size": 0x100},
{ "begin": 0x100000, "tag_carveout_region": [0xffffffff08000,
0xffffffff0Bfff], "block_size": 0x2000}]}
```

Only one of `mte_tag_carveout_json` and `mte_tag_carveout_json_file` can be used.

Type: string

Default value: ""

### **mte\_tag\_carveout\_tag\_order**

Order of the tags within the MTE tag carveout blocks. This can be little-endian (same order as the corresponding tagged data) or big-endian (reverse order).parameter accepts both '-' and '\_', so 'little-endian', 'big-endian', 'little\_endian' and 'big\_endian' are all valid.PARAMETER HAS NO FUNCTIONALITY AT THE MOMENT.

Type: string

Default value: "little-endian"

### pa\_regions\_with\_metadata\_storage

Specify the address region where the metadata storage is available for each PAS in a JSON format.

If the PAS does not have a region specified, the PAS has metadata storage for all of the space.

The regions are defined by begin and end\_incl addresses. Example:

```
{ "ns": [0xa0000000, 0xa0000fff],
  "s"  : [0xb0000000, 0xb0000fff],
  "rl" : [0xc0000000, 0xc0000fff],
  "rt" : [0xd0000000, 0xd0000fff] }
```

ns: non-secure, s: secure, rl: realm, rt: root.

Type: string

Default value: ""

## 3.331 PVWriteBuffer

Defined in `LISA/PVWriteBuffer.lisa`.

### About PVWriteBuffer

The PVWriteBuffer subcomponent buffers PVBus transactions.

### Iris and MTI instances for PVWriteBuffer

This model has the following Iris instances:

Name	Instance type
PVWriteBuffer	PVWriteBuffer
PVWriteBuffer.mapper	PVBusMapper

This model has the following MTI trace components:

Name	Component type
PVWriteBuffer	PVWriteBuffer
PVWriteBuffer.mapper	PVBusMapper

### Ports for PVWriteBuffer

Port	Direction	Protocol	Description
barrier_notify_s	slave	PVWriteBuffer_BarrierPort	Barrier notification input.
clk_in	slave	ClockSignal	Clock input.
pvbus_m	master	PVBus	Master connection to memory bus.
pvbus_s	slave	PVBus	Slave connection for transactions to be buffered.

Port	Direction	Protocol	Description
reset_in	slave	Signal	Reset input.
serror_notify_m	master	PVWriteBuffer_SErrorPort	SError output generation.

## Parameters for PVWriteBuffer

### **buffer\_lifetime**

Natural lifetime (cycles) for data in the write buffer before draining naturally.

Type: `uint32_t`

Default value: 100000

### **number\_of\_regions**

Number of address regions to track.

Type: `unsigned`

Default value: 1867

### **number\_of\_temporal\_buckets**

Number of data buckets to keep.

Type: `unsigned`

Default value: 16

### **obey\_nE\_hint**

Obey the nE (no early return) attribute on incoming transactions.

Type: `bool`

Default value: `false`

## 3.332 PartialWriteDetector

Defined in `LISA/PartialWriteDetector.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

## About PartialWriteDetector

Partial Write Detector for RSE.

## Iris and MTI instances for PartialWriteDetector

This model has the following Iris instances:

Name	Instance type
PartialWriteDetector	PartialWriteDetector

## Ports for PartialWriteDetector

Port	Direction	Protocol	Description
parwrite_addr_out	master	Value	-
parwrite_event_out	master	Signal	-
parwrite_irq_out	master	StateSignal	-
pvbuss_m	master	PVBus	-
pvbuss_s	slave	PVBus	-
pwd_reset_in	slave	Signal	-

## Parameters for PartialWriteDetector

### diagnostics

Diagnostics.

Type: `int32_t`

Default value: 0

### enable\_partial\_write\_detection

Enable Partial Write Detection.

Type: `bool`

Default value: `false`

## 3.333 PchannelListener

Defined in `LISA/PChannelListener.lisa`.

## About PchannelListener

Provides a dummy PChannel device to accept all request.

## Iris and MTI instances for PchannelListener

This model has the following Iris instances:

Name	Instance type
PchannelListener	PChannelListener

### Ports for PchannelListener

Port	Direction	Protocol	Description
dev_pchannel_s	slave	PChannel	-

### Parameters for PchannelListener

#### diagnostics

Diagnostics.

Type: uint32\_t

Default value: 0

## 3.334 PowerStateGate

Defined in LISA/PowerStateGate.lisa.

### About PowerStateGate

Power State Gate to filter the access to SYSTOP domain.

### Iris and MTI instances for PowerStateGate

This model has the following Iris instances:

Name	Instance type
PowerStateGate	PowerStateGate
PowerStateGate.filter	PVBusMapper

This model has the following MTI trace components:

Name	Component type
PowerStateGate.filter	PVBusMapper

### Ports for PowerStateGate

Port	Direction	Protocol	Description
powerdown	slave	Signal	-
pvbus_m	master	PVBus	-
pvbus_s	slave	PVBus	-
reset	slave	Signal	-

## Parameters for PowerStateGate

### **diagnostics**

Diagnostics.

Type: `uint32_t`

Default value: 0

### **domain\_name**

Domain name.

Type: `string`

Default value: "SYSTOP"

### **gate\_behaviour**

Gate behaviour when power is down, 0=abort, 1=ignore.

Type: `uint32_t`

Default value: 0

## 3.335 RAMDevice

Defined in `LISA/RAMDevice.lisa`.

### About RAMDevice

As a generic device, this component does not have a hardware revision code.

### Iris and MTI instances for RAMDevice

This model has the following Iris instances:

Name	Instance type
<code>RAMDevice</code>	<a href="#">RAMDevice</a>
<code>RAMDevice.bus_slave</code>	<a href="#">PVBusSlave</a>

This model has the following MTI trace components:

Name	Component type
<code>RAMDevice.bus_slave</code>	<a href="#">PVBusSlave</a>



## Ports for RAMDevice

Port	Direction	Protocol	Description
pvbus	slave	PVBus	-

## Parameters for RAMDevice

### **enable\_atomic\_ops**

Supports Atomic Operations.

Type: `bool`

Default value: `false`

### **fill1**

Fill pattern 1, initialise memory at start of simulation with alternating fill1, fill2 pattern.

Type: `uint32_t`

Default value: `0xdfdfdfcf`

### **fill2**

Fill pattern 2, initialise memory at start of simulation with alternating fill1, fill2 pattern.

Type: `uint32_t`

Default value: `0xcfdfdfdf`

### **read\_latency**

Memory read latency (ps/byte).

Type: `uint64_t`

Default value: `0`

### **size**

Memory Size.

Type: `uint64_t`

Default value: `0x100000000`

### **write\_latency**

Memory write latency (ps/byte).

Type: `uint64_t`

Default value: `0`

## 3.336 RAM\_ECC\_Checker

Defined in `LISA/RAM_ECC_Checker.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
1	Alpha support

For an explanation of the quality levels, see [Quality level definitions](#).

### About RAM\_ECC\_Checker

RAM ECC Checker.

### Iris and MTI instances for RAM\_ECC\_Checker

This model has the following Iris instances:

Name	Instance type
RAM_ECC_Checker	RAM_ECC_Checker

### Ports for RAM\_ECC\_Checker

Port	Direction	Protocol	Description
double_bit_ecc_addr_out	master	Value	Address for which 2 bit ECC was detected
double_bit_ecc_error_out	master	Signal	-
pvbus_m	master	PVBus	To forward access to external attached memory
pvbus_s	slave	PVBus	To intercept access to external attached memory
reset_in	slave	Signal	Reset Port
single_bit_ecc_addr_out	master	Value	Address for which 1 bit ECC was detected
single_bit_ecc_error_out	master	Signal	-

### Parameters for RAM\_ECC\_Checker

#### diagnostics

Diagnostics.

Type: `uint8_t`

Default value: 2

#### enable\_2bit\_error

When set to true, 2-bits un-recoverable error is reported, else 1-bit error is reported, `enable_ecc_check` must be set.

Type: `bool`

Default value: `false`

### **`enable_ecc_check`**

Enables 1-bit or 2-bits ECC error reporting. When SRAM is preloaded this must be set to false.

Type: `bool`

Default value: `false`

## 3.337 ROM

Defined in `LISA/ROM.lisa`.

### About ROM

Simple ROM device.

### Iris and MTI instances for ROM

This model has the following Iris instances:

Name	Instance type
ROM	ROM
ROM.bus_mapper	PVBusMapper
ROM.bus_slave	PVBusSlave

This model has the following MTI trace components:

Name	Component type
ROM.bus_mapper	PVBusMapper
ROM.bus_slave	PVBusSlave

### Ports for ROM

Port	Direction	Protocol	Description
parity_error_out	master	Signal	-
pvbuss	slave	PVBus	-

### Parameters for ROM

#### **`abort_writes`**

Abort writes instead of ignoring them.

Type: `bool`

Default value: `false`

**log2\_size**

Log2 size (bytes) e.g. 20 is 1 MiB.

Type: unsigned

Default value: 20

**parity\_enabled**

Parity Check Enabled on ROM data: If this parameter is enabled, Model assumes that ROM binary will have data + parity.

Type: bool

Default value: false

**raw\_image**

Raw image file to load at init time.

Type: string

Default value: ""

3.338 RSE\_CPU\_Private\_Region

Defined in LISA/RSE\_CPU\_Private\_Region.lisa.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
1.46	Alpha support

For an explanation of the quality levels, see [Quality level definitions](#).

About RSE\_CPU\_Private\_Region

RSE CPU processor private region.

Iris and MTI instances for RSE\_CPU\_Private\_Region

This model has the following Iris instances:

Name	Instance type
RSE_CPU_Private_Region	<a href="#">RSE_CPU_Private_Region</a>
RSE_CPU_Private_Region.apb_nonsecure	<a href="#">PVBusSlave</a>
RSE_CPU_Private_Region.apb_secure	<a href="#">PVBusSlave</a>

This model has the following MTI trace components:

Name	Component type
RSE_CPU_Private_Region.apb_nonsecure	PVBusSlave
RSE_CPU_Private_Region.apb_secure	PVBusSlave

### Ports for RSE\_CPU\_Private\_Region

Port	Direction	Protocol	Description
apb_nonsecure	slave	PVBus	-
apb_secure	slave	PVBus	secure & non-secure Subordinate APB Interface
reset_in	slave	Signal	Reset in signal

### Parameters for RSE\_CPU\_Private\_Region

#### CPUID\_RESET\_VALUE

CPUID register reset value.

Type: uint32\_t

Default value: 0

#### diagnostics

Diagnostics.

Type: uint32\_t

Default value: 2

## 3.339 RSE\_Integ\_Regs

Defined in LISA/RSE\_Integ\_Regs.lisa.

### About RSE\_Integ\_Regs

RSE Integration Layer Registers.

### Iris and MTI instances for RSE\_Integ\_Regs

This model has the following Iris instances:

Name	Instance type
RSE_Integ_Regs	RSE_Integration_Registers
RSE_Integ_Regs.ClockDivider	ClockDivider
RSE_Integ_Regs.PVBusSlave	PVBusSlave

This model has the following MTI trace components:

Name	Component type
RSE_Integ_Regs.ClockDivider	<a href="#">ClockDivider</a>
RSE_Integ_Regs.PVBusSlave	<a href="#">PVBusSlave</a>

### Ports for RSE\_Integ\_Regs

Port	Direction	Protocol	Description
EXTMCPRESETn	master	<a href="#">Signal</a>	-
EXTSCPRESETn	master	<a href="#">Signal</a>	-
MCP_ATU_AP	master	<a href="#">Signal</a>	-
MCP_RAS_ERR_CLEAR	master	<a href="#">Signal</a>	-
pvbuss_s	slave	<a href="#">PVBus</a>	-
REFCLK	slave	<a href="#">ClockSignal</a>	-
reset_in	slave	<a href="#">Signal</a>	-
RSECORECLK	master	<a href="#">ClockSignal</a>	-
SCP_ATU_AP	master	<a href="#">Signal</a>	-
SCP_RAS_ERR_CLEAR	master	<a href="#">Signal</a>	-
SYSPLLCLK	slave	<a href="#">ClockSignal</a>	-

### Parameters for RSE\_Integ\_Regs

#### **chip\_id**

Chip Identifier.

Type: `uint8_t`

Default value: `0x0`

#### **diagnostics**

Diagnostics.

Type: `uint32_t`

Default value: `0`

#### **multichip\_mode**

Multichip Mode.

Type: `"bool"`

Default value: `false`

## 3.340 RSE\_SystemControl

Defined in `LISA/RSE_SystemControl.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following ports were added:

- `po_reset_aon_out`

### About RSE\_SystemControl

RSE System Control Registers.

### Iris and MTI instances for RSE\_SystemControl

This model has the following Iris instances:

Name	Instance type
<code>RSE_SystemControl</code>	<code>RSE_SystemControl</code>
<code>RSE_SystemControl.busmaster</code>	<a href="#">PVBusMaster</a>
<code>RSE_SystemControl.busslave</code>	<a href="#">PVBusSlave</a>
<code>RSE_SystemControl.scp_rom_busmapper</code>	<a href="#">PVBusMapper</a>

This model has the following MTI trace components:

Name	Component type
<code>RSE_SystemControl.busmaster</code>	<a href="#">PVBusMaster</a>
<code>RSE_SystemControl.busslave</code>	<a href="#">PVBusSlave</a>
<code>RSE_SystemControl.scp_rom_busmapper</code>	<a href="#">PVBusMapper</a>

### Ports for RSE\_SystemControl

Port	Direction	Protocol	Description
<code>boot_addr_out</code>	master	<a href="#">Value_64</a>	Address when <code>boot_en</code> is enabled
<code>boot_en_out</code>	master	<a href="#">Signal</a>	Enables channel 0 to load first command after reset from <code>boot_addr</code>
<code>boot_memattr_out</code>	master	<a href="#">Value</a>	Memory attribute setting for the <code>boot_addr</code>
<code>boot_shareattr_out</code>	master	<a href="#">Value</a>	Shareability attribute for the <code>boot_attr</code>
<code>busmaster_control</code>	master	<a href="#">PVTransactionMaster</a>	-
<code>cpu0_lockup_reset_request</code>	slave	<a href="#">Signal</a>	-
<code>cpu0_warm_reset_request</code>	slave	<a href="#">Signal</a>	-

Port	Direction	Protocol	Description
cpu1_lockup_reset_request	slave	Signal	-
cpu1_warm_reset_request	slave	Signal	-
cpu2_lockup_reset_request	slave	Signal	-
cpu2_warm_reset_request	slave	Signal	-
cpu3_lockup_reset_request	slave	Signal	-
cpu3_warm_reset_request	slave	Signal	-
cpuwait_out	master	Signal	-
dbgen_in	slave	Signal	-
dbgen_out	master	Signal	-
host_level_reset_request	slave	Signal	-
initsvtor	master	Value	-
lcm_dcu_force_disable_out	master	Value	LCM DCU Force disable signal
lcm_reset_request	slave	Signal	-
lcm_sp_reset	master	Signal	-
niden_in	slave	Signal	-
niden_out	master	Signal	-
nonsecure_watchdog_reset_request	slave	Signal	-
pdc_m_pvbus_m	master	PVBus	-
po_reset_aon_out	master	Signal	-
po_reset	master	Signal	-
privileged_access_en_out	master	Signal	Enables DMA privileged access generation during DMA_ICs sequence run
pvbus_s	slave	PVBus	-
reset_in	slave	Signal	-
RESETREQ_in	slave	StateSignal	-
RSE_PSI_STATUS	master	Signal	-
sam_reset_request	slave	Signal	-
scp_cpu_reset	master	Signal	-
scp_rom_access_pvbus_m	master	PVBus	-
scp_rom_access_pvbus_s	slave	PVBus	-
secure_watchdog_reset_request	slave	Signal	-
slow_clock_watchdog_reset_request	slave	Signal	-
software_reset_req_out	master	Signal	SWRESET (PO_RESET) from Primary Chip to Secondary Chip
software_reset_request	slave	Signal	-
spiden_in	slave	Signal	-
spiden_out	master	Signal	-
spniden_in	slave	Signal	-
spniden_out	master	Signal	-



Port	Direction	Protocol	Description
subsystem hardware reset_request	slave	Signal	-
warm_reset	master	Signal	-

## Parameters for RSE\_SystemControl

### COLDRESET\_MODE

Note: The external agents like BMC or ICU or reset controller can be used to reset the RSE in real hardware but in FVP we don't support these external agents Hence can't support full functionality of COLDRESET\_MODE=1 So keeping COLDRESET\_MODE default value to 0.

Type: `uint32_t`

Default value: `0`

### CPU0RSTREQENRST

CPU 0 Warm Reset Request Enable Default Value.

Type: `bool`

Default value: `false`

### CPU0WAITRST

Note: If CPU0WAITRST is changed to `0x1`, then RSE has to make use of DMA boot-flow and DMA has to release the RSE CPU wait signal, this is not currently implemented in FVP. So keeping CPU0WAITRST default value to `false(0)`.

Type: `bool`

Default value: `false`

### CPU1WAITRST

Whether to hold cpu1 in reset at boot.

Type: `bool`

Default value: `true`

### CPU2WAITRST

Whether to hold cpu2 in reset at boot.

Type: `bool`

Default value: `true`

**CPU3WAITRST**

Whether to hold cpu3 in reset at boot.

Type: `bool`

Default value: `true`

**DMA\_BOOT\_EN\_REG\_RESET**

Default Reset value of DMA\_BOOT\_EN register.

Type: `uint32_t`

Default value: `0x0`

**GRETRREG\_RESET**

GRETRREG Reset value.

Type: `uint32_t`

Default value: `0`

**LCM\_DCU\_FORCE\_DISABLE\_REG\_RESET**

Default Reset value of LCM\_DCU\_FORCE\_DISABLE register.

Type: `uint32_t`

Default value: `0x55555555`

**NUMCPU**

Number of Cortex-M CPU cores in the subsystem.

Type: `uint8_t`

Default value: `1`

**NUMDMACHANNEL**

Number of DMA channels.

Type: `uint8_t`

Default value: `2`

**NUMVMBANK**

Number of Volatile Memory Banks.

Type: `uint8_t`

Default value: `2`

**RESET\_SYNDROME\_INIT\_VAL**

Initial value of the RESET\_SYNDROME register.

Type: `uint32_t`

Default value: `0x1`

**RSE\_DMA\_BOOT\_ADDR**

[25:0] bits of this parameter are mapped to bits [27:2] of DMA boot\_addr signal and the DMA\_BOOT\_ADDR register.

Type: `uint32_t`

Default value: `0x407c00`

**RSE\_DMA\_BOOT\_REGION**

[3:0] bits of this parameter are mapped to bits [31:28] of DMA boot\_addr signal.

Type: `uint8_t`

Default value: `0x1`

**SWRESETREQ\_BIT**

Software Reset Request Bit.

Type: `uint8_t`

Default value: `5`

**allow\_lockup\_mask**

Whether to allow masking of CPU lockup reset.

Type: `bool`

Default value: `true`

**diagnostics**

Diagnostics.

Type: `int32_t`

Default value: `0`

**reset\_vector\_addr**

Reset Vector Address.

Type: `uint32_t`

Default value: 0x11000000

### 3.341 RandomNumberGenerator

Defined in `LISA/RandomNumberGenerator.lisa`.

#### About RandomNumberGenerator

Random Number Generator unit.

#### Iris and MTI instances for RandomNumberGenerator

This model has the following Iris instances:

Name	Instance type
RandomNumberGenerator	<a href="#">RandomNumberGenerator</a>
RandomNumberGenerator.pvbusslave	<a href="#">PVBusSlave</a>

This model has the following MTI trace components:

Name	Component type
RandomNumberGenerator.pvbusslave	<a href="#">PVBusSlave</a>

#### Ports for RandomNumberGenerator

Port	Direction	Protocol	Description
pvbus_s	slave	<a href="#">PVBus</a>	Bus slave interface.
RNG_intr	master	<a href="#">Signal</a>	Interrupt output.

#### Parameters for RandomNumberGenerator

##### diagnostics

Diagnostics.

Type: `uint32_t`

Default value: 0

##### seed

Random number seed.

Type: `uint32_t`

Default value: 0

## 3.342 RealTimeLimiter

Defined in `examples/LISA/Common/LISA/RealTimeLimiter.lisa`.

### About RealTimeLimiter

Real Time Limiter.

### Iris and MTI instances for RealTimeLimiter

This model has the following Iris instances:

Name	Instance type
<code>RealTimeLimiter</code>	<code>RealTimeLimiter</code>
<code>RealTimeLimiter.divider</code>	<code>ClockDivider</code>

This model has the following MTI trace components:

Name	Component type
<code>RealTimeLimiter.divider</code>	<code>ClockDivider</code>

### Ports for RealTimeLimiter

Port	Direction	Protocol	Description
<code>clk_in</code>	slave	<code>ClockSignal</code>	Clock input.

### Parameters for RealTimeLimiter

#### **ENABLE**

Rate limit simulation.

Type: `bool`

Default value: `false`

#### **RELATIVE\_SPEED**

Rate limit to at most this percentage of real time (100: limit to wall clock rate).

Type: `int`

Default value: 100

## 3.343 RealtimeClockTimer

Defined in `LISA/RealtimeClockTimer.lisa`.

### About RealtimeClockTimer

Host Time Based Timer Module for Generic Timers.

### Ports for RealtimeClockTimer

Port	Direction	Protocol	Description
set_frequency	slave	Value_64	-
timer_callback	master	TimerCallback64	-
timer_control	slave	TimerControl64	-

### Parameters for RealtimeClockTimer

This component does not have any parameters.

## 3.344 RemapDecoder

Defined in `examples/LISA/Common/LISA/RemapDecoder.lisa`.

### About RemapDecoder

The component that provides support for dynamically remappable regions of memory.

### Iris and MTI instances for RemapDecoder

This model has the following Iris instances:

Name	Instance type
RemapDecoder	RemapDecoder
RemapDecoder.bus_switch	TZSwitch
RemapDecoder.bus_switch.pvbus_mapper	PVBusMapper

This model has the following MTI trace components:

Name	Component type
RemapDecoder.bus_switch.pvbus_mapper	PVBusMapper

### Ports for RemapDecoder

Port	Direction	Protocol	Description
control	broadcast	TZSwitchControl	-
input	slave	PVBus	Incoming bus transactions (connected straight to TZSwitch).
output_remap_clear	master	PVBus	Outgoing bus transactions when remap is clear.
output_remap_set	master	PVBus	Outgoing bus transactions when remap is set.

Port	Direction	Protocol	Description
remap	slave	StateSignal	Remapping control.

### Parameters for RemapDecoder

This component does not have any parameters.

## 3.345 RootKeyStorage

Defined in `LISA/RootKeyStorage.lisa`.

### About RootKeyStorage

Trusted Root-Key Storage unit.

### Iris and MTI instances for RootKeyStorage

This model has the following Iris instances:

Name	Instance type
RootKeyStorage	RootKeyStorage
RootKeyStorage.pvbusslave	PVBusSlave

This model has the following MTI trace components:

Name	Component type
RootKeyStorage.pvbusslave	PVBusSlave

### Ports for RootKeyStorage

Port	Direction	Protocol	Description
pdbus_s	slave	PVBus	-

### Parameters for RootKeyStorage

#### **diagnostics**

Diagnostics.

Type: `uint32_t`

Default value: 0

#### **hw\_unique\_key**

Hardware Unique Key (128-bit, 4 std::hex words).

Type: `string`

Default value: "00000000 00000000 00000000 00000000"

**hw\_unique\_key\_hex**

Hardware Unique Key (128-bit, little-endian std::hex byte stream).

Type: `string`

Default value: `N/A`

**private\_key**

Private Endorsement Key (256-bit, 8 std::hex words).

Type: `string`

Default value: `"00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000"`

**private\_key\_hex**

Private Key (256-bit, little-endian std::hex byte stream).

Type: `string`

Default value: `N/A`

**public\_key**

Public Key (256-bit, 8 std::hex words).

Type: `string`

Default value: `"00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000"`

**public\_key\_hex**

Public Key (256-bit, little-endian std::hex byte stream).

Type: `string`

Default value: `N/A`

**ss\_key**

Secret Symmetric Key (128-bit, 4 std::hex words).

Type: `string`

Default value: `"00000000 00000000 00000000 00000000"`

**ss\_key\_hex**

Secret Symmetric Key (128-bit, little-endian std::hex byte stream).

Type: `string`

Default value: `N/A`



**version**

Version of the model functionality. Valid values are r0 and r1.

Type: string

Default value: "r1"

## 3.346 SC\_ClockSignal2ClockSignal

Defined in `examples/SystemCEExport/Bridges/SC_ClockSignal2ClockSignal.lisa`.

### About SC\_ClockSignal2ClockSignal

SystemC ClockSignal to ClockSignal converter.

### Iris and MTI instances for SC\_ClockSignal2ClockSignal

This model has the following Iris instances:

Name	Instance type
SC_ClockSignal2ClockSignal	SC_ClockSignal2ClockSignal

### Ports for SC\_ClockSignal2ClockSignal

Port	Direction	Protocol	Description
clk_in	master	ClockSignal	-
sc_clk_in	slave	SC_ClockSignal	-

### Parameters for SC\_ClockSignal2ClockSignal

This component does not have any parameters.

## 3.347 SDC600

Defined in `LISA/SDC600.lisa`.

### About SDC600

CoreSight Secure Debug Channel component.

### Iris and MTI instances for SDC600

This model has the following Iris instances:

Name	Instance type
SDC600	SDC600

## Ports for SDC600

Port	Direction	Protocol	Description
apb_s	slave	PVBus	For register read and write access
irq_out	master	Signal	Interrupt output
reset_in	slave	Signal	Reset input
sdcm_com_pvbuss_m	master	PVBus	Port to connect to Cortex-M DAP.

## Parameters for SDC600

### **diagnostics**

Diagnostics.

Type: uint8\_t

Default value: 2

### **m\_core\_address**

M core base address to store the certificate.

Type: uint32\_t

Default value: 0x0

### **pidr0\_reg\_reset\_value**

PIDR0 Register reset value.

Type: uint32\_t

Default value: 0xB9

### **pidr1\_reg\_reset\_value**

PIDR1 Register reset value.

Type: uint32\_t

Default value: 0xEF

### **pidr2\_reg\_reset\_value**

PIDR2 Register reset value.

Type: uint32\_t

Default value: 0x0B

### **pidr4\_reg\_reset\_value**

PIDR4 Register reset value.

Type: `uint32_t`

Default value: `0x04`

### **romentry\_reg\_value**

Reset value of ROMENTRY register, used to point to the ROM table of the target system.

Type: `uint32_t`

Default value: `0`

### **rx\_buf\_size**

RX buffer size.

Type: `uint32_t`

Default value: `1`

### **sr\_reg\_reset\_value**

SR Register reset value.

Type: `uint32_t`

Default value: `0x80001001`

### **system\_type**

System type. 0:ADiv6 1:ADiv5.2 2:Cortex-M DAP - DEFAULT==0.

Type: `uint8_t`

Default value: `0`

### **tx\_buf\_size**

TX buffer size.

Type: `uint32_t`

Default value: `1`

## 3.348 SGSignal2AMBAPVSignal

Defined in `examples/SystemCEExport/Bridges/SGSignal2AMBAPVSignal.lisa`.

### About SGSignal2AMBAPVSignal

SGSignal to AMBA-PV signal protocol converter.



Variants of this component also exist with multiple input and output ports.

### Iris and MTI instances for SGSignal2AMBAPVSignal

This model has the following Iris instances:

Name	Instance type
SGSignal2AMBAPVSignal	SGSignal2AMBAPVSignal

### Ports for SGSignal2AMBAPVSignal

Port	Direction	Protocol	Description
amba_pv_signal_m	master	AMBAPVSignal	Output master port for connection to top-level AMBAPVSignal master port. Converted signal state changes are sent out through this port.
sg_signal_s	slave	Signal	Handles incoming signal state changes.

### Parameters for SGSignal2AMBAPVSignal

This component does not have any parameters.

## 3.349 SGSignal2AMBAPVSignalx16

Defined in `examples/SystemCEExport/Bridges/SGSignal2AMBAPVSignalx16.lisa`.

### About SGSignal2AMBAPVSignalx16

SGSignal to AMBA-PV Signal protocol converter



Variants of this component also exist with multiple input and output ports.

### Iris and MTI instances for SGSignal2AMBAPVSignalx16

This model has the following Iris instances:

Name	Instance type
SGSignal2AMBAPVSignalx16	SGSignal2AMBAPVSignalx16

### Ports for SGSignal2AMBAPVSignalx16

Port	Direction	Protocol	Description
amba_pv_signal_m	master	AMBAPVSignal	Output master port for connection to top-level AMBAPVSignal master port. Converted signal state changes are sent out through this port.

Port	Direction	Protocol	Description
sg_signal_s	slave	Signal	Handles incoming signal state changes.

### Parameters for SGSignal2AMBAPVSignalx16

This component does not have any parameters.

## 3.350 SGSignal2AMBAPVSignalx224

Defined in `examples/SystemCEExport/Bridges/SGSignal2AMBAPVSignalx224.lisa`.

### About SGSignal2AMBAPVSignalx224

SGSignal to AMBA-PV Signal protocol converter



Variants of this component also exist with multiple input and output ports.

### Iris and MTI instances for SGSignal2AMBAPVSignalx224

This model has the following Iris instances:

Name	Instance type
SGSignal2AMBAPVSignalx224	SGSignal2AMBAPVSignalx224

### Ports for SGSignal2AMBAPVSignalx224

Port	Direction	Protocol	Description
amba_pv_signal_m	master	AMBAPVSignal	Output master port for connection to top-level AMBAPVSignal master port. Converted signal state changes are sent out through this port.
sg_signal_s	slave	Signal	Handles incoming signal state changes.

### Parameters for SGSignal2AMBAPVSignalx224

This component does not have any parameters.

## 3.351 SGSignal2AMBAPVSignalx256

Defined in `examples/SystemCEExport/Bridges/SGSignal2AMBAPVSignalx256.lisa`.

### About SGSignal2AMBAPVSignalx256

SGSignal to AMBA-PV Signal protocol converter



Variants of this component also exist with multiple input and output ports.

### Iris and MTI instances for SGSignal2AMBAPVSignalx256

This model has the following Iris instances:

Name	Instance type
SGSignal2AMBAPVSignalx256	SGSignal2AMBAPVSignalx256

### Ports for SGSignal2AMBAPVSignalx256

Port	Direction	Protocol	Description
amba_pv_signal_m	master	AMBAPVSignal	Output master port for connection to top-level AMBAPVSignal master port. Converted signal state changes are sent out through this port.
sg_signal_s	slave	Signal	Handles incoming signal state changes.

### Parameters for SGSignal2AMBAPVSignalx256

This component does not have any parameters.

## 3.352 SGSignal2AMBAPVSignalx4

Defined in `examples/SystemCEExport/Bridges/SGSignal2AMBAPVSignalx4.lisa`.

### About SGSignal2AMBAPVSignalx4

SGSignal to AMBA-PV Signal protocol converter



Variants of this component also exist with multiple input and output ports.

### Iris and MTI instances for SGSignal2AMBAPVSignalx4

This model has the following Iris instances:

Name	Instance type
SGSignal2AMBAPVSignalx4	SGSignal2AMBAPVSignalx4

### Ports for SGSignal2AMBAPVSignalx4

Port	Direction	Protocol	Description
amba_pv_signal_m	master	AMBAPVSignal	Output master port for connection to top-level AMBAPVSignal master port. Converted signal state changes are sent out through this port.

Port	Direction	Protocol	Description
sg_signal_s	slave	Signal	Handles incoming signal state changes.

### Parameters for SGSignal2AMBAPVSignalx4

This component does not have any parameters.

## 3.353 SGSignal2AMBAPVSignalx48

Defined in `examples/SystemCEExport/Bridges/SGSignal2AMBAPVSignalx48.lisa`.

### About SGSignal2AMBAPVSignalx48

SGSignal to AMBA-PV Signal protocol converter



Variants of this component also exist with multiple input and output ports.

### Iris and MTI instances for SGSignal2AMBAPVSignalx48

This model has the following Iris instances:

Name	Instance type
SGSignal2AMBAPVSignalx48	SGSignal2AMBAPVSignalx48

### Ports for SGSignal2AMBAPVSignalx48

Port	Direction	Protocol	Description
amba_pv_signal_m	master	AMBAPVSignal	Output master port for connection to top-level AMBAPVSignal master port. Converted signal state changes are sent out through this port.
sg_signal_s	slave	Signal	Handles incoming signal state changes.

### Parameters for SGSignal2AMBAPVSignalx48

This component does not have any parameters.

## 3.354 SGSignal2AMBAPVSignalx8

Defined in `examples/SystemCEExport/Bridges/SGSignal2AMBAPVSignalx8.lisa`.

### About SGSignal2AMBAPVSignalx8

SGSignal to AMBA-PV Signal protocol converter



Variants of this component also exist with multiple input and output ports.

### Iris and MTI instances for SGSignal2AMBAPVSignalx8

This model has the following Iris instances:

Name	Instance type
SGSignal2AMBAPVSignalx8	SGSignal2AMBAPVSignalx8

### Ports for SGSignal2AMBAPVSignalx8

Port	Direction	Protocol	Description
amba_pv_signal_m	master	AMBAPVSignal	Output master port for connection to top-level AMBAPVSignal master port. Converted signal state changes are sent out through this port.
sg_signal_s	slave	Signal	Handles incoming signal state changes.

### Parameters for SGSignal2AMBAPVSignalx8

This component does not have any parameters.

## 3.355 SGSignalBuffer

Defined in `LISA/SGSignalBuffer.lisa`.

### About SGSignalBuffer

This component buffers changes to its input signal, outputting the new values at the next clock tick. This is useful in SystemC export situations as the output signal is driven from an `SC_THREAD` whereas the input could have come from an `SC_METHOD`. This avoids issues if a downstream component calls `sc_wait()`.



There are variants of the buffer with different port array sizes.

### Iris and MTI instances for SGSignalBuffer

This model has the following Iris instances:

Name	Instance type
SGSignalBuffer	SGSignalBuffer



## Ports for SGSignalBuffer

Port	Direction	Protocol	Description
clk_in	slave	ClockSignal	Clock rate to release buffered signals.
in	slave	Signal	Signal in.
out	master	Signal	Buffered signal out.

## Parameters for SGSignalBuffer

This component does not have any parameters.

## 3.356 SGSignalBufferx16

Defined in `LISA/SGSignalBufferx16.lisa`.

### About SGSignalBufferx16

This component buffers changes to its input signal, outputting the new values at the next clock tick. This is useful in SystemC export situations as the output signal is driven from an `SC_THREAD` whereas the input could have come from an `SC_METHOD`. This avoids issues if a downstream component calls `sc_wait()`.



There are variants of the buffer with different port array sizes.

### Iris and MTI instances for SGSignalBufferx16

This model has the following Iris instances:

Name	Instance type
SGSignalBufferx16	SGSignalBufferx16

## Ports for SGSignalBufferx16

Port	Direction	Protocol	Description
clk_in	slave	ClockSignal	Clock rate to release buffered signals.
ins	slave	Signal	Signal in.
outs	master	Signal	Buffered signal out.

## Parameters for SGSignalBufferx16

This component does not have any parameters.

### 3.357 SGSignalBufferx2

Defined in `LISA/SGSignalBufferx2.lisa`.

#### About SGSignalBufferx2

This component buffers changes to its input signal, outputting the values at the next clock tick. This is useful in SystemC export situations as the output signal is driven from an `SC_THREAD` whereas the input could have come from an `SC_METHOD`. This avoids issues if a downstream component calls `sc_wait()`.



There are variants of the buffer with different port array sizes.

#### Iris and MTI instances for SGSignalBufferx2

This model has the following Iris instances:

Name	Instance type
SGSignalBufferx2	SGSignalBufferx2

#### Ports for SGSignalBufferx2

Port	Direction	Protocol	Description
clk_in	slave	ClockSignal	Clock rate to release buffered signals.
ins	slave	Signal	Signal in.
outs	master	Signal	Buffered signal out.

#### Parameters for SGSignalBufferx2

This component does not have any parameters.

### 3.358 SGSignalBufferx4

Defined in `LISA/SGSignalBufferx4.lisa`.

#### About SGSignalBufferx4

This component buffers changes to its input signal, outputting the values at the next clock tick. This is useful in SystemC export situations as the output signal is driven from an `SC_THREAD` whereas the input could have come from an `SC_METHOD`. This avoids issues if a downstream component calls `sc_wait()`.



There are variants of the buffer with different port array sizes.

### Iris and MTI instances for SGSignalBufferx4

This model has the following Iris instances:

Name	Instance type
SGSignalBufferx4	SGSignalBufferx4

### Ports for SGSignalBufferx4

Port	Direction	Protocol	Description
clk_in	slave	ClockSignal	Clock rate to release buffered signals.
ins	slave	Signal	Signal in.
outs	master	Signal	Buffered signal out.

### Parameters for SGSignalBufferx4

This component does not have any parameters.

## 3.359 SGSignalBufferx8

Defined in LISA/SGSignalBufferx8.lisa.

### About SGSignalBufferx8

This component buffers changes to its input signal, outputting the new values at the next clock tick. This is useful in SystemC export situations as the output signal is driven from an SC\_THREAD whereas the input could have come from an SC\_METHOD. This avoids issues if a downstream component calls `sc_wait()`.



There are variants of the buffer with different port array sizes.

### Iris and MTI instances for SGSignalBufferx8

This model has the following Iris instances:

Name	Instance type
SGSignalBufferx8	SGSignalBufferx8

## Ports for SGSignalBufferx8

Port	Direction	Protocol	Description
clk_in	slave	ClockSignal	Clock rate to release buffered signals.
ins	slave	Signal	Signal in.
outs	master	Signal	Buffered signal out.

## Parameters for SGSignalBufferx8

This component does not have any parameters.

## 3.360 SGSignalBufferx988

Defined in LISA/SGSignalBufferx988.lisa.

## About SGSignalBufferx988

This component buffers changes to its input signal, outputting the new values at the next clock tick. This is useful in SystemC export situations as the output signal is driven from an SC\_THREAD whereas the input could have come from an SC\_METHOD. This avoids issues if a downstream component calls `sc_wait()`.



Note

There are variants of the buffer with different port array sizes.

## Iris and MTI instances for SGSignalBufferx988

This model has the following Iris instances:

Name	Instance type
SGSignalBufferx988	SGSignalBufferx988

## Ports for SGSignalBufferx988

Port	Direction	Protocol	Description
clk_in	slave	ClockSignal	Clock rate to release buffered signals.
ins	slave	Signal	Signal in.
outs	master	Signal	Buffered signal out.

## Parameters for SGSignalBufferx988

This component does not have any parameters.

### 3.361 SGStateSignal2AMBAPVSignalState

Defined in `examples/SystemCEExport/Bridges/SGStateSignal2AMBAPVSignalState.lisa`.

#### About SGStateSignal2AMBAPVSignalState

SGStateSignal to AMBA-PV SignalState protocol converter.



Variants of this component also exist with multiple input and output ports.

#### Iris and MTI instances for SGStateSignal2AMBAPVSignalState

This model has the following Iris instances:

Name	Instance type
SGStateSignal2AMBAPVSignalState	SGStateSignal2AMBAPVSignalState

#### Ports for SGStateSignal2AMBAPVSignalState

Port	Direction	Protocol	Description
amba_pv_signal_m	master	AMBAPVSignalState	Output master port for connection to top-level AMBAPVValue64 master port. Converted value changes are sent out through this port.
sg_signal_s	slave	StateSignal	Handles incoming value changes.

#### Parameters for SGStateSignal2AMBAPVSignalState

This component does not have any parameters.

### 3.362 SGStateSignal2AMBAPVSignalStatex4

Defined in `examples/SystemCEExport/Bridges/SGStateSignal2AMBAPVSignalStatex4.lisa`.

#### About SGStateSignal2AMBAPVSignalStatex4

SGStateSignal to AMBA-PV SignalState protocol converter.



Variants of this component also exist with multiple input and output ports.

#### Iris and MTI instances for SGStateSignal2AMBAPVSignalStatex4

This model has the following Iris instances:

Name	Instance type
SGStateSignal2AMBAPVSignalStatex4	SGStateSignal2AMBAPVSignalStatex4

### Ports for SGStateSignal2AMBAPVSignalStatex4

Port	Direction	Protocol	Description
amba_pv_signal_m	master	AMBAPVSignalState	Output master port for connection to top-level AMBAPVValue64 master port. Converted value changes are sent out through this port.
sg_signal_s	slave	StateSignal	Handles incoming value changes.

### Parameters for SGStateSignal2AMBAPVSignalStatex4

This component does not have any parameters.

## 3.363 SGValue2AMBAPVValue

Defined in `examples/SystemCEExport/Bridges/SGValue2AMBAPVValue.lisa`.

### About SGValue2AMBAPVValue



Variants of this component also exist with multiple input and output ports.

### Iris and MTI instances for SGValue2AMBAPVValue

This model has the following Iris instances:

Name	Instance type
SGValue2AMBAPVValue	SGValue2AMBAPVValue

### Ports for SGValue2AMBAPVValue

Port	Direction	Protocol	Description
amba_pv_value_m	master	AMBAPVValue	Output master port for connection to top-level AMBAPVValue master port. Converted value changes are sent out through this port.
sg_value_s	slave	Value	Handles incoming value changes.

### Parameters for SGValue2AMBAPVValue

This component does not have any parameters.

## 3.364 SGValue2AMBAPVValue64

Defined in `examples/SystemCEExport/Bridges/SGValue2AMBAPVValue64.lisa`.

### About SGValue2AMBAPVValue64



Variants of this component also exist with multiple input and output ports.

### Iris and MTI instances for SGValue2AMBAPVValue64

This model has the following Iris instances:

Name	Instance type
SGValue2AMBAPVValue64	SGValue2AMBAPVValue64

### Ports for SGValue2AMBAPVValue64

Port	Direction	Protocol	Description
amba_pv_value_m	master	AMBAPVValue64	Output master port for connection to top-level AMBAPVValue master port. Converted value changes are sent out through this port.
sg_value_s	slave	Value_64	Handles incoming value changes.

### Parameters for SGValue2AMBAPVValue64

This component does not have any parameters.

## 3.365 SGValue2AMBAPVValue64x4

Defined in `examples/SystemCEExport/Bridges/SGValue2AMBAPVValue64x4.lisa`.

### About SGValue2AMBAPVValue64x4

Value\_64 to AMBA-PV Value64 protocol converter



Variants of this component also exist with multiple input and output ports.

### Iris and MTI instances for SGValue2AMBAPVValue64x4

This model has the following Iris instances:

Name	Instance type
SGValue2AMBAPVValue64x4	SGValue2AMBAPVValue64x4

### Ports for SGValue2AMBAPVValue64x4

Port	Direction	Protocol	Description
amba_pv_value_m	master	AMBAPVValue64	Output master port for connection to top-level AMBAPVValue master port. Converted value changes are sent out through this port.
sg_value_s	slave	Value_64	Handles incoming value changes.

### Parameters for SGValue2AMBAPVValue64x4

This component does not have any parameters.

## 3.366 SGValue2AMBAPVValue4

Defined in `examples/SystemCEExport/Bridges/SGValue2AMBAPVValue4.lisa`.

### About SGValue2AMBAPVValue4

Value to AMBA-PV Value protocol converter



Variants of this component also exist with multiple input and output ports.

### Iris and MTI instances for SGValue2AMBAPVValue4

This model has the following Iris instances:

Name	Instance type
SGValue2AMBAPVValue4	SGValue2AMBAPVValue4

### Ports for SGValue2AMBAPVValue4

Port	Direction	Protocol	Description
amba_pv_value_m	master	AMBAPVValue	Output master port for connection to top-level AMBAPVValue master port. Converted value changes are sent out through this port.
sg_value_s	slave	Value	Handles incoming value changes.

### Parameters for SGValue2AMBAPVValue4

This component does not have any parameters.



## 3.367 SGValueState2AMBAPVValueState

Defined in `examples/SystemCEExport/Bridges/SGValueState2AMBAPVValueState.lisa`.

### About SGValueState2AMBAPVValueState



Variants of this component also exist with multiple input and output ports.

### Iris and MTI instances for SGValueState2AMBAPVValueState

This model has the following Iris instances:

Name	Instance type
SGValueState2AMBAPVValueState	SGValueState2AMBAPVValueState

### Ports for SGValueState2AMBAPVValueState

Port	Direction	Protocol	Description
amba_pv_value_m	master	AMBAPVValueState	Output master port for connection to top-level AMBAPVValue master port. Converted value changes are sent out through this port.
sg_value_s	slave	ValueState	Handles incoming value changes.

### Parameters for SGValueState2AMBAPVValueState

This component does not have any parameters.

## 3.368 SGValueState2AMBAPVValueState64

Defined in `examples/SystemCEExport/Bridges/SGValueState2AMBAPVValueState64.lisa`.

### About SGValueState2AMBAPVValueState64



Variants of this component also exist with multiple input and output ports.

### Iris and MTI instances for SGValueState2AMBAPVValueState64

This model has the following Iris instances:

Name	Instance type
SGValueState2AMBAPVValueState64	SGValueState2AMBAPVValueState64

### Ports for SGValueState2AMBAPVValueState64

Port	Direction	Protocol	Description
amba_pv_value_m	master	AMBAPVValueState64	Output master port for connection to top-level AMBAPVValue64 master port. Converted value changes are sent out through this port.
sg_value_s	slave	ValueState_64	Handles incoming value changes.

### Parameters for SGValueState2AMBAPVValueState64

This component does not have any parameters.

## 3.369 SGValueState2AMBAPVValueState64x4

Defined in `examples/SystemCEExport/Bridges/SGValueState2AMBAPVValueState64x4.lisa`.

### About SGValueState2AMBAPVValueState64x4

ValueState\_64 to AMBA-PV ValueState64 protocol converter



Note

Variants of this component also exist with multiple input and output ports.

### Iris and MTI instances for SGValueState2AMBAPVValueState64x4

This model has the following Iris instances:

Name	Instance type
SGValueState2AMBAPVValueState64x4	SGValueState2AMBAPVValueState64x4

### Ports for SGValueState2AMBAPVValueState64x4

Port	Direction	Protocol	Description
amba_pv_value_m	master	AMBAPVValueState64	Output master port for connection to top-level AMBAPVValue64 master port. Converted value changes are sent out through this port.
sg_value_s	slave	ValueState_64	Handles incoming value changes.

### Parameters for SGValueState2AMBAPVValueState64x4

This component does not have any parameters.

## 3.370 SGValueState2AMBAPVValueStatex4

Defined in `examples/SystemCEExport/Bridges/SGValueState2AMBAPVValueStatex4.lisa`.

### About SGValueState2AMBAPVValueStatex4

ValueState to AMBA-PV ValueState protocol converter



Variants of this component also exist with multiple input and output ports.

### Iris and MTI instances for SGValueState2AMBAPVValueStatex4

This model has the following Iris instances:

Name	Instance type
SGValueState2AMBAPVValueStatex4	SGValueState2AMBAPVValueStatex4

### Ports for SGValueState2AMBAPVValueStatex4

Port	Direction	Protocol	Description
amba_pv_value_m	master	AMBAPVValueState	Output master port for connection to top-level AMBAPVValue master port. Converted value changes are sent out through this port.
sg_value_s	slave	ValueState	Handles incoming value changes.

### Parameters for SGValueState2AMBAPVValueStatex4

This component does not have any parameters.

## 3.371 SI\_L1

Defined in `LISA/SI_L1.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Preliminary support

For an explanation of the quality levels, see [Quality level definitions](#).

### About SI\_L1

- Major IP revisions (rX) are modeled and are controlled by the model `revision` parameter.
- Minor IP revisions (pY) are not modeled. There are no functional differences in the model between pY revisions, with the exception of the `peripheral_id2` register.

- To configure the model, you must have installed Arm Socrates. The model's `mesh_config_file` parameter defines the network interfaces present as well as their configuration options. Set it to the name of the YAML configuration file emitted by the Socrates export process. Fast Models requires the configuration file to pass Design Rule Check (DRC) in Socrates and does not support manually editing the configuration file. You must use version r1p8-01 or later of Socrates to generate the file. For more information about Socrates, see [Arm Socrates](#) on Arm Developer. To download it, contact Arm Technical Support.
- Interconnect models do not typically model RTL defects.
- The mapping between the port number for ASNI, AMNI, HSNi, HMNI, and PMNI ports and the xxNI name used in the yml file is its index when all xxNIs of that same type are sorted in ascending alphabetical order. For example, if the yml file has three ASNIs `asni_s100_scp`, `asni_s101_dap`, and `asni_s204_periph0`:
  - `asni_s100_scp` is mapped to `pvbuss_s_asni[0]`
  - `asni_s101_dap` is mapped to `pvbuss_s_asni[1]`
  - `asni_s204_periph0` is mapped to `pvbuss_s_asni[2]`

Similarly:

- AMNIs are mapped to `pvbuss_m_amni`
- HSNIs are mapped to `pvbuss_s_hsn`
- HMNIs are mapped to `pvbuss_m_hmni`
- PMNIs are mapped to `pvbuss_m_pmni`

Additionally, the name-to-index mappings are printed when the component parameter `print_parser_log=true`.



The PMNI RTL supports up to 16 external interfaces, however the model uses a single PMNI port. Through the single PMNI port, the models support targeting each of the 16 interfaces independently in the SAM.



- Access to a non-existent register inside of a CFGNI address space is **WI/RAZ**. This commonly happens when accessing a unused register in an endpoint due to a feature being disabled.
- Access to a non-existent register outside of a CFGNI address space is DECERR. This commonly happens when accessing a location that is outside of the rendered config pages, but inside of the config space declared inside of the yaml. Note that the configuration space can be defined in the yaml larger than what is required for the registers.

Address map-related support:

- Model supports Static Maps and Programmable Address Maps (PAM).
  - Static Map hashed and non-hashed memory regions are parsed from the `mesh_config_file`.

- (PAM flow) Support for default\_tgt\_id strap to set default xSNI targets.
- Supports configurable address mask (cmp\_addr\_mask\_{l,u}, htg\_addr\_mask\_{l,u}).
- Model supports topology parameter sam/regionCompLSB to set minimum SAM address granule with values between 4KB and 64KB.
- Model supports No\_Target as a target in Address Maps.
- Model supports remap states in static map flow.

Programmer's view related support:

- Model supports 4KB and 64KB for topology parameter configNodeGranularity.
- Model supports discovery to determine the system address of the following node registers:
  - Global, Voltage, Power, and Clock Domain registers.
  - ASNI, HSNI, AMNI, HMNI, PMNI, and PMU registers.
  - IDM, APU, and PAM subfeature registers.
  - FMU and FCU.

MCN related support:

- Up to 8 MCNs can be added in a specific configuration.
- Each MCN only exposes one upstream Coherent Domain Interface and one downstream Memory Controller Interface.
- Supports discovery process with partial support for MCN. See MCN limitations.
- MCN registers can be accessed via any subordinate interfaces (xSNI such as ASNI). See the Limitations section for the location of the MCN registers' address space.
- MCN programmers view functionality (registers with behavior modeled):
  - MCN Secure Access (MCN\_SCR)
  - MCN Architecture State (arch\_state\_req, arch\_state\_cur)
- Each MCN has a System Level Cache inside it.



Unless any feature is listed here as supported, it should be assumed that it is absent from the model.

---

## Changes since 11.30

- Registers refreshed, except MCN MSF, to match RTL rOp0 LAC release. No new functionality modeled and below are the major registers changes:
  - AMNI: added a NETWORK\_INFO register and weighted arbitration controls.
  - ASNI: added three new UID mask registers.
  - CMNI: AUX\_CONTROL was moved to offset 0x00000810, after RDDATA\_AGG\_CONTROL.
  - Global: added NODE\_INFO.

- PwrDomain: Every IDM power-domain error register has been renamed from `IDM_PD_ERROR_*` to `IDM_PD_ERR_*`
- MCN:
  - Removed “N” or “M” suffix where there is a multiple number of a given register.
  - AMU: added `AMDEVARCH`.
  - GCB: added interrupt registers (`INTENCLR/SET`, `INTRR`, `INTSCLR/SET`, `OVSCCLR/SET`, and `CBUSY_CTRL` for secure only).
  - MPAM: added `MPAMCFG_CMIN`. Removed implementation defined (MPAM RISO and RIS1) `SLCCFG*` and `RPMSMON` registers.
  - RAS: added `ERR3MISC1` and `ERR3MISC2` registers.
- Documented pre-existing limitation: No support for RME registers, root and realm register present but mapped to secure access.

## Limitations



Note

Issues listed in this section have identifiers of the form [SDDKW-x]. These are for Arm internal references only.

- The following features are out of scope for the Fast Model, and will not be supported:
  - PMU counters are not supported. Counter registers are implemented as **RAZ**.
  - QoS is not supported and all related registers are **RAZ/WI**.
  - Power, clock and interrupt signals are not supported.
  - No support for AHB Locked transfers.
  - No support for IDM timeout detection.
  - No support for reorder buffers or Cyclic Dependency Avoidance Scheme (CDAS).
- APU limitations [SDDKW-82596]:
  - The base and size of address regions must be aligned to 4KB even when `apuRegion4k` is false. A warning will be given and the region may not function properly due to PVBUS limitations.
  - `APU_ENABLE_RESET_STRAP` is not supported and `APU_CTRL.apu_enable` bit should be used to enable APU instead.
  - APU Region Locking is currently not supported. [SDDKW-80598]
  - `APUID` is not read from the transaction attributes, instead the model parameter `apu_subsystem_id` is used to specify the `subsystemID/APUID` of the requestor sending in a transaction through an `xSNI`. [SDDKW-81014]
  - If a transaction comes through an `xSNI` which does not have an APU and is routed to an `xMNI` which does have one, the APU in the `xMNI` treats the `APUID` as 0.

- Since the `subsystemID/APUID` is not encoded in the transaction, components downstream to this model can't know the APUID of the requestor.
- Address map and bus traffic limitations [SDDKW-82595]:
  - Default target id cannot be configured through `sam_status` register. [SDDKW-77767] It can be provided through `default_tgtid_strap_i` model parameter.
  - Routers are not modeled. `xSNI` access only considers whether the target is defined in the `xSNI`'s memory map. [SDDKW-79760]
  - No support for AMNI Address shuttering. [SDDKW-80419]
  - Model supports 1, 2, and 4 targets for power-of-two stripe group hashing. No other striping functions/target combinations are supported.
  - Hashing of stripe groups limited to a minimum granularity of 4096B. This is a DMI limitation of PVBUS. See [Bus traffic in Fast Models](#) for information.
  - No support for exclusive monitoring (ASNI/HSNI/AMNI/HMNI). [SDDKW-79385]
  - `xSNI` access to `CFGNI` is not limited by router connectivity defined in the `mesh_config_file`. It considers only whether the `xSNI` has the `CFGNI` target defined in its memory map.
- Stripe limitations
  - ASNI striping to an HMNI target has not been tested.
  - Limited testing of stripe groups with different numbers of targets and granularities.
  - No support for single target stripe.
  - There is no support for additional granularities, or the additional stripe group remap functionality.
  - Limited 8-way striping testing for static address maps. Not supported for programmable address maps.
  - User-defined stripe function is not supported for programmable address maps.
- Remapping features not supported:
  - Tested 5 out of the maximum of 8 remap states.
  - The priority for multiple address remapping states.
  - One target remapped to a different target.
  - A single target remapped to a stripe group.
  - One stripe group remapped to a different stripe group.
  - A stripe group remapped to a single target.
- Programmer's view limitations [SDDKW-82591]:
  - 64-bit register accesses are not supported. [SDDKW-75304]
  - No support for secure/root override for register accesses. [SDDKW-77158]
  - IDM registers are not tested. [SDDKW-77474]
  - No support for `idm_sreset_strap_i` functionality. [SDDKW-80451]

- AMNI nodes in the model are interface-indifferent and the registers may not reflect the protocol version.
- Error injection and error generation (RAS) are not supported. All error registers are **RAZ/WI**. [SDDKW-73411]
- No support for MTE.
- Register bitfields with W1S are treated as RW.
- ASNI `read_channel_mpam_override` and `write_channel_mpam_override` and the HSNI `MPAM_CONTROL` register fields `*mpam_override_value` are limited to 11 bits. [SDDKW-102675]
- No support for RME registers, root and realm register present but mapped to secure access.
- Topology size limitations [SDDKW-82589]:
  - Maximum number of Voltage Domains is 32. The maximum tested is 1.
  - Maximum number of Power Domains is 32. The maximum tested is 2.
  - Maximum number of Clock Domains is 32. The maximum tested is 16.
  - Maximum number of PMU nodes is 32 (1 per Clock Domain). The maximum tested is 15.
  - Maximum number of Subordinate Network Interfaces (SNIs) is 128 (ASNI + HSNI). The maximum tested is 128 ASNIs and 9 HSNI.
  - Maximum number of Manager Network Interfaces (MNIs) is 127 (AMNI + HMNI + PMNI). The maximum tested is 127 AMNIs, 127 PMNIs, and 9 HMNIs.
- FMU and FCU limitations:
  - Registers are readable and writeable. No other functionality is modeled.
- CMO limitations:
  - No software control for the CMO terminate response in register `AMNI_CONFIG_CTL`.
  - CMO Response control is not supported.
- A warm reset using static address maps has not been tested. [SDDKW-84005]
- AMNI nodes in the model are interface-indifferent and registers do not reflect the protocol version.
- A reset after model startup does not reset the registers or address remap selections.
- No register visibility support for a debugger.
- AXI-Stream (TSNI and TMNI) is not supported
- AMNI and HMNI exclusive monitors are not supported
- Ordered Write Observation (OWO) is not supported
- Resource Planes (RP) are not supported
- Unique TX id generator is not supported
- More than 32 Clock Domains are not supported
- [SDDKW-82598] DVMs initiated by the DSUs that enter the MCN via the Coherent Domain Interface are not propagated upstream.



- Component Aggregation Layer (CAL2) is not supported.
- MCN limitations:
  - The `NODE_ID` field in the discovery `NODE_TYPE` register reads as 0.
  - No support for de-striping and configurable address truncation.
  - Snoop support not tested.
  - Interrupt interface is not supported.
  - Memory Side Filter (MSF) has not been tested.
  - SLC does not cache data.
  - OCM is not supported.
  - [SDDKW-98048] MCN register offsets do not match the RTL. There are 3 workarounds:
    - The MCN register offset can be found through discovery.
    - The parameter `print_config` shows the addresses of the model register blocks.
    - The TRM section “Configuration register address region calculation” contains a formula for calculating the register region size. Removing the MCN part of the calculation determines the model starting location of the first MCN.
  - Data flow is not suppressed when MCN is in the `CONFIG` architecture state.
  - Iris
    - No debugger support for reading or setting parameters.
    - MSF registers incorrectly show up in two groups. Use the MSF group instead of the ones in the MCN group.
- Limited CMNI support for the MCNs:
  - Configuration-dependent register reset values have not been tested.
  - CMNI subfeature discovery has not been tested.

## Iris and MTI instances for SI\_L1

This model has the following Iris instances:

Name	Instance type
<code>SI_L1</code>	<code>SI_L1</code>
<code>SI_L1.decoder</code>	<code>PVBusMapper</code>
<code>SI_L1.dsu_mapper</code>	<code>PVBusMapper</code>
<code>SI_L1.mcnY</code> (where $Y = 0-3$ )	<code>MCN</code>
<code>SI_L1.mcnY.bus_mapper</code> (where $Y = 0-3$ )	<code>PVBusMapper</code>
<code>SI_L1.mcnY.cdi_logger</code> (where $Y = 0-3$ )	<code>PVBusLogger</code>
<code>SI_L1.mcnY.cdi_logger.mapper</code> (where $Y = 0-3$ )	<code>PVBusMapper</code>
<code>SI_L1.mcnY.exclusive_monitor</code> (where $Y = 0-3$ )	<code>PVBusExclusiveMonitor</code>
<code>SI_L1.mcnY.exclusive_monitor.bus_mapper</code> (where $Y = 0-3$ )	<code>PVBusMapper</code>
<code>SI_L1.mcnY.memory_if_logger</code> (where $Y = 0-3$ )	<code>PVBusLogger</code>
<code>SI_L1.mcnY.memory_if_logger.mapper</code> (where $Y = 0-3$ )	<code>PVBusMapper</code>

Name	Instance type
SI_L1.mcnY.metadata_controller (where Y = 0-3)	PVMetadataController
SI_L1.mcnY.metadata_controller.MetaDataMapper (where Y = 0-3)	PVBusMapper
SI_L1.mcnY.metadata_controller.mdc_downstream_port[0].pvbusmaster (where Y = 0-3)	PVBusMaster
SI_L1.mcnY.msf (where Y = 0-3)	MSF
SI_L1.mcnY.msf.mapper_cdi (where Y = 0-3)	PVBusMapper
SI_L1.mcnY.msf.mapper_ncdi (where Y = 0-3)	PVBusMapper
SI_L1.mcnY.msf.pvbus_slave (where Y = 0-3)	PVBusSlave
SI_L1.mcnY.ncdi_logger (where Y = 0-3)	PVBusLogger
SI_L1.mcnY.ncdi_logger.mapper (where Y = 0-3)	PVBusMapper
SI_L1.mcnY.programming_if_logger (where Y = 0-3)	PVBusLogger
SI_L1.mcnY.programming_if_logger.mapper (where Y = 0-3)	PVBusMapper
SI_L1.mcnY.system_level_cache.cache (where Y = 0-3)	PVCache
SI_L1.warning_memory	WarningMemory

This model has the following MTI trace components:

Name	Component type
SI_L1	SI_L1
SI_L1.decoder	PVBusMapper
SI_L1.dsu_mapper	PVBusMapper
SI_L1.mcnY.bus_mapper (where Y = 0-3)	PVBusMapper
SI_L1.mcnY.cdi_logger (where Y = 0-3)	PVBusLogger
SI_L1.mcnY.cdi_logger.mapper (where Y = 0-3)	PVBusMapper
SI_L1.mcnY.exclusive_monitor (where Y = 0-3)	PVBusExclusiveMonitor
SI_L1.mcnY.exclusive_monitor.bus_mapper (where Y = 0-3)	PVBusMapper
SI_L1.mcnY.memory_if_logger (where Y = 0-3)	PVBusLogger
SI_L1.mcnY.memory_if_logger.mapper (where Y = 0-3)	PVBusMapper
SI_L1.mcnY.metadata_controller (where Y = 0-3)	MetaDataController
SI_L1.mcnY.metadata_controller.MetaDataMapper (where Y = 0-3)	PVBusMapper
SI_L1.mcnY.metadata_controller.mdc_downstream_port[0].pvbusmaster (where Y = 0-3)	PVBusMaster
SI_L1.mcnY.msf.mapper_cdi (where Y = 0-3)	PVBusMapper
SI_L1.mcnY.msf.mapper_ncdi (where Y = 0-3)	PVBusMapper
SI_L1.mcnY.msf.pvbus_slave (where Y = 0-3)	PVBusSlave
SI_L1.mcnY.ncdi_logger (where Y = 0-3)	PVBusLogger
SI_L1.mcnY.ncdi_logger.mapper (where Y = 0-3)	PVBusMapper
SI_L1.mcnY.programming_if_logger (where Y = 0-3)	PVBusLogger
SI_L1.mcnY.programming_if_logger.mapper (where Y = 0-3)	PVBusMapper

## Ports for SI\_L1

Port	Direction	Protocol	Description
idm_reset_signal_amni	master	Signal	IDM reset signals to AMNIs.

Port	Direction	Protocol	Description
idm_reset_signal_asni	master	Signal	IDM reset signals to ASNI.
idm_reset_signal_hmni	master	Signal	IDM reset signals to HMNI.
idm_reset_signal_hsni	master	Signal	IDM reset signals to HSNIs.
idm_reset_signal_pmni	master	Signal	IDM reset signals to PMNI.
mcn_excl_mon_cleared	master	Signal	Clear events for the MCN exclusive monitors.
pvbus_m_amni	master	PVBus	AMNI downstream ports.
pvbus_m_hmni	master	PVBus	HMNI downstream ports.
pvbus_m_mem_if	master	PVBus	MCN downstream memory interface ports.
pvbus_m_pmni	master	PVBus	PMNI downstream ports.
pvbus_s_asni	slave	PVBus	ASNI upstream ports.
pvbus_s_dsus_if	slave	PVBus	MCN DSU interface (CDI) ports.
pvbus_s_hsni	slave	PVBus	HSNI upstream ports.
reset_in	slave	Signal	Reset signal.

## Parameters for SI\_L1

### **apu\_subsystem\_id**

APUID/SubsystemID of the component connected to each xSNI.

Specify the subsystem id of the component connected to each <x>SNI by using a format like:

```
<x>SNI<m>=<subsystemID0>,<y>SNI<n>=<subsystemID1>
```

The subsystemID is assumed to be 0 for any component connected to an <x>SNI that does not appear in this list.

Type: string

Default value: ""

### **bypass\_msf**

If true, MCN will bypass the Memory Side Filter(MSF).

Type: bool

Default value: true

### **default\_tgt\_id\_strap\_i**

Default Target ID input.

Specify the target id for each <x>SNI by using a format like:

```
<x>SNI<m>=<target_id0>,<y>SNI<n>=<target_id1>
```

CFGNI (Configuration Network Interface) is used as the default target if a certain <x>SNI does not appear in the list.

Type: `string`

Default value: `""`

### **disable\_dsu\_striping**

Whether to disable address striping on `pvbus_s_dsu_if` port 0.

Set to `true` (default) if the interconnect does not perform striping on data from the `pvbus_s_dsu_if` ports (all `pvbus_s_dsu_if` ports are used). Set to `false` if the interconnect will perform striping of data from `pvbus_s_dsu_if` port 0 to `pvbus_m_mem_if` ports 0 to N (`pvbus_s_dsu_if` ports 1 to N are unused).

Type: `bool`

Default value: `true`

### **mesh\_config\_file**

Name of a file containing mesh placement of `SI_L1` components.

Type: `string`

Default value: `""`

### **mpam\_attributes**

User-defined transform to be applied to bus attributes like `ManagerID`, `ExtendedID` Or `UserFlags`, for MPAM Attributes encoded into bus attributes.

For example:

```
ExtendedID[62:55]=MPAM_PMG, ExtendedID[54:39]=MPAM_PARTID, ExtendedID[38]=MPAM_NS
```

An empty string disables MPAM support.

Type: `string`

Default value: `""`

### **periphbase**

Value for `PERIPHBASE`.

Type: `uint64_t`

Default value: `0xffffffffffffffff`

**print\_config**

Enables printing the config register addresses.

Type: `bool`

Default value: `false`

**print\_parser\_log**

Enables printing the yaml config parser log messages.

Type: `bool`

Default value: `false`

**revision**

Component revision. Currently supports r0p0.

Type: `string`

Default value: `"r0p0"`

**show\_banner**

Show component banner:

**0**

suppress entire banner

**1**

suppress config file

**2+**

show full banner.

Type: `uint64_t`

Default value: 2

## 3.372 SI\_System\_Ctrl\_Regs

Defined in `LISA/SI_System_Ctrl_Regs.lisa`.

### About SI\_System\_Ctrl\_Regs

Safety Island System Control Registers.

### Iris and MTI instances for SI\_System\_Ctrl\_Regs

This model has the following Iris instances:

Name	Instance type
SI_System_Ctrl_Regs	SI_System_Control_Registers
SI_System_Ctrl_Regs.pvbusslave	PVBusSlave

This model has the following MTI trace components:

Name	Component type
SI_System_Ctrl_Regs.pvbusslave	PVBusSlave

### Ports for SI\_System\_Ctrl\_Regs

Port	Direction	Protocol	Description
cpuhalt_m	master	Signal	-
pvbus_s	slave	PVBus	-
reset_in	slave	Signal	-
rvbar_cl0	master	Value_64	-
rvbar_cl1	master	Value_64	-
rvbar_cl2	master	Value_64	-

### Parameters for SI\_System\_Ctrl\_Regs

#### **cl0\_c0\_cfgrvbaraddr**

CL0\_CO\_CFGRVBARADDR.

Type: uint64\_t

Default value: 0x120000000

#### **cl1\_c0\_cfgrvbaraddr**

CL1\_CO\_CFGRVBARADDR.

Type: uint64\_t

Default value: 0x140000000

#### **cl1\_c1\_cfgrvbaraddr**

CL1\_C1\_CFGRVBARADDR.

Type: uint64\_t

Default value: 0x140002000

#### **cl2\_c0\_cfgrvbaraddr**

CL2\_CO\_CFGRVBARADDR.

Type: uint64\_t

Default value: 0x160000000

**cl2\_c1\_cfgrvbaraddr**

CL2\_C1\_CFGRVBARADDR.

Type: uint64\_t

Default value: 0x160002000

**cl2\_c2\_cfgrvbaraddr**

CL2\_C2\_CFGRVBARADDR.

Type: uint64\_t

Default value: 0x160004000

**cl2\_c3\_cfgrvbaraddr**

CL2\_C3\_CFGRVBARADDR.

Type: uint64\_t

Default value: 0x160008000

**cpuhalt\_reset**

CPU HALT Reset value.

Type: uint32\_t

Default value: 0

**diagnostics**

Diagnostics.

Type: uint32\_t

Default value: 0

## 3.373 SMCF

Defined in `LISA/SMCF.lisa`.

### About SMCF

System Monitoring Control Framework (SMCF).

### Iris and MTI instances for SMCF

This model has the following Iris instances:

Name	Instance type
SMCF	SMCF

## Ports for SMCF

Port	Direction	Protocol	Description
mgi_clk_in	slave	ClockSignal	Clock input
mgi_irq_out	master	Signal	Interrupt signal output
mgi_pvbus_m	master	PVBus	For DMA or memory mapped data write
mgi_reg_pvbus_s	slave	PVBus	To access MGI register
mgi_reset_in	slave	Signal	Reset signal input
mgi_tag_in	slave	Value	Tag value from external hardware
mgi_trigger_in	slave	Signal	To trigger the start of a sample by external hardware
mgi_trigger_out	master	Signal	Signal to external hardware to indicate that an event has occurred in an MGI
mli_hsp_enable_ack_in	slave	Signal	Acknowledgement signal for enable signal in hsp.
mli_hsp_enable_out	master	Signal	Signal to enable Ring Oscillators in hsp.
mli_powerdown_in	slave	Signal	It gives info about AP is power OFF or not
mli_temperature_in	slave	ValueState	It will be connected to temperature sensor to fetch the temperature value
smcf_mli_pvbus_m	master	PVBus	manager port to read/write the AMU or HSP register value

## Parameters for SMCF

### **ALERT\_NUM\_CFG**

Specifies the number of alerts present A value of 0 means that no alerts are present.

Type: uint8\_t

Default value: 0

### **ALT\_ADDR\_CFG**

Specifies where monitor data is read from, it is either: 0: The MGI\_DATA<n> registers. 1: The address specified in MGI\_RADDR0/1.

Type: bool

Default value: 0

### **ALT\_DELTA\_CFG**

Specifies the presence of alert rising and falling delta functions 0: The rising and falling delta functions are not present 1: The rising and falling delta functions are present.

Type: bool

Default value: 0



**DATA\_PER\_MON\_CFG**

Specifies the number of data values(DATA\_PER\_MON\_CFG+1) generated from each monitor.

Type: uint32\_t

Default value: 0

**DEF\_CFG\_IRQ\_MASK**

Specifies the default value of the configuration interrupt event mask. Sets the reset value of MGI\_IRQ\_MASK.CFG\_IRQ\_MASK.

Type: bool

Default value: 1

**DEF\_CFG\_TRIG\_MASK**

Specifies the default value of the configuration trigger event mask. Sets the reset value of MGI\_TRG\_MASK.CFG\_TRIG\_MASK.

Type: bool

Default value: 1

**DEF\_RADDR\_CFG**

Specifies the default alternate read address. This sets the default value for MGI\_RADDR0/MGI\_RADDR1.

Type: uint64\_t

Default value: 0x0

**DEF\_WADDR\_CFG**

Specifies the default write address for the DMA interface. This sets the default value for MGI\_WADDR0/MGI\_WADDR1. Only required if DMA\_IF\_CFG = 1. This value must be 32-bit aligned.

Type: uint64\_t

Default value: 0x0

**DMA\_IF\_CFG**

Specifies the presence of the DMA interface 0: DMA interface is not present 1: DMA interface is present.

Type: bool

Default value: 0

**END\_COMPONENT**

Based on this parameter value sampling value will be fetched from the respective model 0: Temperature sensor 1: Monitor unit (AMU) 2: Fake sensor/monitor 3: HSP (Hot Spot Profiler) 4: DTSV2(Distributed Thermal Sensor V2) 5: DTS(Digital Distributed Temperature Sensor).

Type: uint8\_t

Default value: 0

**FAKE\_SENSOR\_MAX\_LIMIT**

Maximum value limit for the fake sensor/monitor.

Type: uint64\_t

Default value: 0xFFFF

**FAKE\_SENSOR\_MIN\_LIMIT**

Minimum value limit for the fake sensor/monitor.

Type: uint64\_t

Default value: 0

**GRP\_ID\_CFG**

Specifies a unique identifier for an MGI.

Type: uint16\_t

Default value: 0

**MGI\_AIDR\_RESET\_VALUE**

Reset value for MGI\_AIDR register.

Type: uint8\_t

Default value: 0x10

**MGI\_IIDR\_RESET\_VALUE**

Reset value for MGI\_IIDR register.

Type: uint32\_t

Default value: 0x8D00043B

**MLI\_QUANTITY**

MLI\_QUANTITY.

Type: `uint8_t`

Default value: 1

### **MODE\_LEN\_CFG**

Specifies the bit width of each MGI\_MODE\_REQ/STAT. The number of bits is MODE\_LEN\_CFG +1.

Type: `uint8_t`

Default value: 31

### **MODE\_REG\_CFG**

Specifies the number of MGI\_MODE\_REQ/STAT pairs. A value of 0 means that no MGI\_MODE\_REQ/STAT pairs are present.

Type: `uint8_t`

Default value: 0

### **MONITOR\_SAMPLE\_TIME**

Time that monitor takes to perform the sampling (essentially the time from MGI sending the sample\_start command to the MGI receiving all the sample data).

Type: `uint32_t`

Default value: 0x13FFF

### **MON\_BASE\_ADDRESS**

The base address is used for sampling and connecting monitors. It consists of the hexadecimal value given in string format. If there are multiple MLI connected then this parameter will have the base addresses of the MLI's in hexadecimal format and separated by comma.

Type: `string`

Default value: "0x00000000"

### **MON\_DATA\_WIDTH\_CFG**

Specifies the bit width(MON\_DATA\_WIDTH\_CFG+1) of each monitor data value.

Type: `uint8_t`

Default value: 31

### **MON\_DISCON\_CFG**

Specifies if each monitor supports being disconnected. For example, bit 0 represents monitor 0, and bit 3 represents monitor 3. 0b0: Monitor does not support being disconnected. It is reset to connected. 0b1: Monitor supports being disconnected. It is reset to disconnected.

Type: `uint32_t`

Default value: `0x0`

### **MON\_NUM\_CFG**

Specifies the number of monitors(`MON_NUM_CFG+1`) in an MGI.

Type: `uint8_t`

Default value: `0`

### **NUM\_OF\_RSP\_CONNECTED**

Number of maximum RSPs can be connected to the DTSV2.

Type: `uint8_t`

Default value: `9`

### **PACKED\_CFG**

Specifies if monitor data is packed 0: Data is not packed 1: Data is packed.

Type: `bool`

Default value: `0`

### **PER\_TIMER\_CFG**

Specifies the presence of the periodic timer 0: The periodic timer is not present 1: The periodic timer is present.

Type: `bool`

Default value: `0`

### **SINGLE\_MON\_MODE\_CFG**

Specifies that each monitor is a single type and all monitors will have the same mode setting.

Type: `bool`

Default value: `0`

### **SMP\_DLY\_LEN\_CFG**

Specifies the bit width of `MGI_SMP_DLY`. The number of bits in `SMP_DLY_LEN_CFG`. A value of 0 means this register is not present and the feature is not supported.

Type: `uint8_t`

Default value: `0`

**TAG\_IN\_CFG**

Specifies the presence of the tag input 0: The tag input is not present 1: The tag input is present.

Type: `bool`

Default value: 1

**TAG\_LEN\_CFG**

Specifies the bit width of the tag input The tag bit width is TAG\_LEN\_CFG+1.

Type: `uint8_t`

Default value: 31

**TRIG\_IN\_CFG**

Specifies the presence of the input trigger interface 0: The trigger in interface is not present 1: The trigger in interface is present.

Type: `bool`

Default value: 1

**TRIG\_OUT\_CFG**

Specifies the presence of the output trigger interface 0: The trigger out interface is not present 1: The trigger out interface is present.

Type: `bool`

Default value: 1

**USER\_DEF\_CMD\_CFG**

Specifies if an MGI supports User-Defined commands. 0: User-Defined commands are not supported. 1: User-Defined commands are supported.

Type: `bool`

Default value: 0

**diagnostics**

Diagnostics.

Type: `uint8_t`

Default value: 2

## 3.374 SMMUv3AEM

Defined in `LISA/SMMUv3AEM.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following parameters were added:

- `NoStreamID_for_unaware_of_RME`
- `dh_cmo_behavior`
- `imp_def_MTCOMB_ATSResponse_N_bit`

### About SMMUv3AEM

The SMMUv3 Architecture Envelope Model component is an architectural model that implements the SMMUv3.0 to SMMUv3.4 architectures for I/O virtualization of devices, except for the limitations listed below.

The SMMUv3 specifies that input addresses are conceptually 64 bits. The SMMUv3AEM model assumes that the input address is 64 bits. If the SoC has less than 64 bits as an input address bus then if the SoC wants to use the high address space (and use TTB1) then it must sign extend the address from the upstream peripherals to get to 64 bits.

### Limitations

- No power control
- AMBA stash operations, destructive read and destructive hint operations are not supported on PVBUS and also are not supported by the device.
- The PMU has limited functionality. Only a subset of the architecturally mandatory events are supported, as indicated by the `SMMU_PMC_G_CBEID0` fields. The PMU is intended for demonstration purposes only and for driver development.



Note

Between 11.17 and 11.18, the point of triggering for events changed for 'TLB miss' and this might lead to an (architecturally valid) change in the values captured in some circumstances.

- It has limited RAS support configured by the `ras` parameter
- PCIe No\_snoop transactions are not supported
- SMMUv3.3 PMCG filtering by MPAM PARTID and PMG is not supported.
- SMMUv3.4 D128 descriptors are not supported.

## Notes

- A bad configuration renders the model inactive.
- Some configurations can be adjusted by configuration pins. These are only sampled at the negative edge of the reset pin. If you want to use these pins then you must drive them before sending a negative edge on the reset pin. During simulation reset the component driving them must also drive this transition again.
- Debug reads to the registers do not disturb the state.
- Writes to registers with Update flags, including debug writes, are ignored if the Update flag is already set to one.
- Debug and real accesses to the registers must be 32 bits or 64 bits.
- The model deals with groups of transactions with the same attributes and a similar range of addresses. The mapping used is remembered by the bus infrastructure and is used for subsequent sufficiently similar transactions without requiring intervention from the SMMU model, and is not traced, for instance.

The range of addresses that the mapping is valid for is determined by the SMMU model, depending on architectural and model-implementation details. However, as it is unaware of any sign-extension or the mapping that the SoC does, the SoC is responsible for subsequently narrowing the range of addresses for which this mapping is valid. Typically, this is done automatically when using PVBUSMapper.

- If the downstream system of the SMMU needs to distinguish the kinds of SMMU-originated accesses then:
  - For SMMU-originated writes, MSIs are issued using attributes determined by the parameter `msi_attribute_transform`, whilst Event queue writes are always issued with `ExtendedID=0`, `UserFlags=0`, `ManagerID64=0xffffffff` unless overridden by the parameter `tw_qs_attribute_transform`.
  - For SMMU-originated reads and compare-and-swap (CAS) operations, `tw_qs_attribute_transform` can be used.
  - If your system does table walks and queue accesses through TBUO (`separate_tw_msi_qs_port == false`), then care must be taken to distinguish table walk and queue traffic from normal translated traffic.
- If `SMMU_IDR1.TABLES_PRESET` or `SMMU_IDR1.QUEUES_PRESET` is set then see parameter `PRESET_REL_base_address` and the parameters it mentions.

Embedded implementations of the SMMU are allowed to have the queues/stream table in a 'close' RAM, either on-chip or in the SMMU itself. For the model, it is up to the integrator to supply this memory and for the SMMU model to be able to access it. Thus if the actual hardware has the memory built into the SMMU then it will be necessary for the integrator to wrap this model with a bus decoder and a memory model to more closely model the embedded implementation.

## Security State Determination (SSD)

The model uses the term SSD to mean the security state that the transaction, register, structure, or event belongs to. In the SMMUv3 architecture, the sideband signal `SEC_SID` holds this information for the transactions, but uses a different encoding.

Before RME, `SEC_SID` was a one-bit signal. With RME, in the hardware, it was expanded to two bits. To retain backwards compatibility in the model, `SEC_SID` remains as one-bit in the parameter `howto_identify`, but the second bit can be expressed with `SEC_SID_bit_1`, and its negative logic version `nSEC_SID_bit_1`.

Security state	SSD	SEC_SID_bit_1	SEC_SID
secure	0	0	1
non-secure	1	0	0
root (reserved)	2	1	1
realm	3	1	0

Parameters for parsing transaction attributes

Some of the parameters are strings that share a common parsing format for extracting from and placing information into the transaction's attributes.

They can extract and place information into the following fields of a transaction's attributes:

- `ManagerID64` (64 bits)
- `ExtendedID` (64 bits)
- `UserFlags` (32 bits)

The string parameter is parsed as a comma-separated list of:

```
lhs_expression=rhs_expression
```

The `lhs_expression` and `rhs_expression` can be an entire symbol, for example:

```
StreamID
```

or a bit, or bit slice:

```
StreamID[16]  
StreamID[31:20]
```

In `rhs_expression`, numeric constants (or slices of numeric constants) are also allowed:

```
StreamID[16]=1  
StreamID[16]=10[2]
```

A single symbol might be assigned from multiple non-contiguous arrays of bits from a mix of different RHS symbols:

```
StreamID[16]=1, StreamID[31:30]=ExtendedID[1:0],  
StreamID[15:0]=UserFlags[31:16], ...
```



In those cases where a left hand symbol can also appear on the right hand side, it is possible to swap bits and transform the symbol. For example, the following expression swaps bits 0 and 1 of the ExtendedID:

```
ExtendedID[0]=ExtendedID[1], ExtendedID[1]=ExtendedID[0]
```

Any bits of the attributes that have no transform specified are retained from the input.

The `lhs_expression` and `rhs_expression` must have the same bit width.

## Iris and MTI instances for SMMUv3AEM

This model has the following Iris instances:

Name	Instance type
SMMUv3AEM	SMMUv3AEM
SMMUv3AEM.axi_stream_msi_manager[4294967295].pvbusmaster	PVBusMaster
SMMUv3AEM.pvbus_id_routed_m[4294967295].pvbusmaster	PVBusMaster
SMMUv3AEM.register_file[0]	PVBusSlave
SMMUv3AEM.service_request_tbu[Y] (where Y = 0-63)	PVBusSlave
SMMUv3AEM.tbu[Y] (where Y = 0-63)	PVBusMapper
SMMUv3AEM.tw_msi_qs_manager[4294967295].pvbusmaster	PVBusMaster

This model has the following MTI trace components:

Name	Component type
SMMUv3AEM	SMMUv3AEM
SMMUv3AEM.axi_stream_msi_manager[4294967295].pvbusmaster	PVBusMaster
SMMUv3AEM.pvbus_id_routed_m[4294967295].pvbusmaster	PVBusMaster
SMMUv3AEM.register_file[0]	PVBusSlave
SMMUv3AEM.service_request_tbu[Y] (where Y = 0-63)	PVBusSlave
SMMUv3AEM.tbu[Y] (where Y = 0-63)	PVBusMapper
SMMUv3AEM.tw_msi_qs_manager[4294967295].pvbusmaster	PVBusMaster

## Ports for SMMUv3AEM

Port	Direction	Protocol	Description
axi_stream_msi_addr_to_match_s	slave	Value_64	Any SMMU-originated MSI which exactly matches the address in this port will be sent through the AXI stream port <code>axi_stream_msi_m</code> which is usually connected to the GIC through <code>axi_stream_msi_s</code> . As MSIs are 32 bit aligned then if bits[1:0] != 0 or the address is above OAS then this is effectively disabled. The parameter <code>smmu_msi_device_id</code> is the DeviceID to send on the interface. See also the parameters: <code>axi_stream_msi_TID</code> and <code>axi_stream_msi_TDEST</code> . The default value of this port is set by the parameter <code>axi_stream_msi_addr_to_match</code> . This port is sampled at negedge of <code>reset_in</code> and must be set before the negedge of the reset signal.

Port	Direction	Protocol	Description
axi_stream_msi_m	master	PVBus	Manager port used for sending SMMU originated MSIs directly to the GIC when axi_stream_msi_enabled == true
clk_in	slave	ClockSignal	Clock signal
conf_reset_of_SMMU_GBPA_ABORT	slave	Signal	System reset value of SMMU_GBPA.ABORT. This port is sampled at negedge of reset_in and must be set before the negedge of the reset signal.
conf_reset_of_SMMU_S_GBPA_ABORT	slave	Signal	System reset value of SMMU_S_GBPA.ABORT. This port is sampled at negedge of reset_in and must be set before the negedge of the reset signal.
conf_system_supports_bgptm	slave	Signal	System supports broadcast TLBI PAALL and TLBI RPA for supporting RME. This port is sampled at negedge of reset_in and must be set before the negedge of the reset signal.
conf_system_supports_btm	slave	Signal	System supports BTM and will be reflected in the IDR registers. This signal can override the value set by the parameters configuring the IDR registers. If BTM (Broadcast Table Maintenance) is not supported then DVM messages will be ignored. This port is sampled at negedge of reset_in and must be set before the negedge of the reset signal.
conf_system_supports_cohacc	slave	Signal	System supports COHACC and will be reflected in the IDR registers. This signal can override the value set by the parameters configuring the IDR registers. If COHACC is set then page walks and SMMU-generated accesses will have the required shareability set, otherwise they will be marked as non-shareable. This port is sampled at negedge of reset_in and must be set before the negedge of the reset signal.
conf_system_supports_httu	slave	Signal	System supports HTTU and will be reflected in the IDR registers. See parameter support_for_httu_when_starts_disallowed for the use of this signal. This port is sampled at negedge of reset_in and must be set before the negedge of the reset signal.
conf_system_supports_sev	slave	Signal	System supports SEV and will be reflected in the IDR registers. This signal can override the value set by the parameters configuring the IDR registers. This port is sampled at negedge of reset_in and must be set before the negedge of the reset signal.
identify	master	SMMUV3AEMIdentifyProtocol	Map the transaction to the tuple (StreamID, SubstreamID, SubstreamIDValid, SSD)
irq_out_command_queue_sync_ns	master	Signal	Pulsed interrupt output signal for non-secure CMD_SYNC having a completion signal of SIG_IRQ.
irq_out_command_queue_sync_rl	master	Signal	Pulsed interrupt output signal for realm CMD_SYNC having a completion signal of SIG_IRQ.
irq_out_command_queue_sync_s	master	Signal	Pulsed interrupt output signal for secure CMD_SYNC having a completion signal of SIG_IRQ.
irq_out_event_queue_ns	master	Signal	Pulsed interrupt output signal for the non-secure event queue becoming non-empty.
irq_out_event_queue_rl	master	Signal	Pulsed interrupt output signal for the realm event queue becoming non-empty.
irq_out_event_queue_s	master	Signal	Pulsed interrupt output signal for the secure event queue becoming non-empty.
irq_out_gerror_ns	master	Signal	Pulsed interrupt output signal for non-secure SMMU_GERROR(N) signaling an error.

Port	Direction	Protocol	Description
irq_out_gerror_rl	master	Signal	Pulsed interrupt output signal for realm SMMU_GERROR(N) signaling an error.
irq_out_gerror_s	master	Signal	Pulsed interrupt output signal for secure SMMU_GERROR(N) signaling an error.
irq_out_gpf_far	master	Signal	For RME-enabled SMMUs. A new error reported in SMMU_ROOT_GPF_FAR will pulse this interrupt.
irq_out_gpt_cfg_far	master	Signal	For RME-enabled SMMUs. A new error reported in SMMU_ROOT_GPT_CFG_FAR will pulse this interrupt.
irq_out_ns	master	Signal	Pulsed interrupt output signal combined from all non-secure (non-RAS) interrupts. This delivers a pulse if any of the interrupt pins of the specified world also deliver a pulse. SW will have to poll all the different reasons to see why it was delivered.
irq_out_pmcg_ns_as_value	master	Value	Non-secure PMCG interrupt value port. Value port representing a set of Performance Monitor Counter Group interrupts. There is an unknown number of PMCGs and so an unknown number of PMCG interrupts. There may not necessarily even be an interrupt per group. We export the interrupt to be generated as a unsigned. The 'pmcg_index' is exported on the top 16 bits and the 'pmcg_counter' (that overflowed) on the bottom 16 bits. Any configured MSI for this group will be the next transaction out of the pvtbus_m_tw_msi_qs port. The architecture supports an MSI from a PMCG could come out of a different port (say the TBU manager port that a PMCG might be associated with). However the AEM only supports it coming out of the TCU port (pvtbus_m_tw_msi_qs).
irq_out_pmcg_s_as_value	master	Value	Secure PMCG interrupt value port. Value port representing a set of Performance Monitor Counter Group interrupts. There is an unknown number of PMCGs and so an unknown number of PMCG interrupts. There may not necessarily even be an interrupt per group. We export the interrupt to be generated as a unsigned. The 'pmcg_index' is exported on the top 16 bits and the 'pmcg_counter' (that overflowed) on the bottom 16 bits. Any configured MSI for this group will be the next transaction out of the pvtbus_m_tw_msi_qs port. The architecture supports an MSI from a PMCG could come out of a different port (say the TBU manager port that a PMCG might be associated with). However the AEM only supports it coming out of the TCU port (pvtbus_m_tw_msi_qs).
irq_out_pri_queue_rl	master	Signal	Pulsed interrupt output signal for the realm PRI queue.
irq_out_pri_queue	master	Signal	Pulsed interrupt output signal for the non-secure PRI queue.
irq_out_ras_cri_as_value	master	Value	RAS Critical error interrupt value port. The number of RAS error record groups is dependent on the configuration. As each error record group could have its own interrupts then we write to this value port the error record group number (which is dependent on the configuration) to simulate an interrupt pulse. We only support modelling edge-triggered interrupts. The value passed is formatted as follows: bits[23:16] the ras error group index (if multiple error groups) (0 atm) bits[15:8] node id bits[7:0] the ras record index in the node.

Port	Direction	Protocol	Description
irq_out_ras_eri_as_value	master	Value	RAS Error Handling Interrupt value port. The number of RAS error record groups is dependent on the configuration. As each error record group could have its own interrupts then we write to this value port the error record group number (which is dependent on the configuration) to simulate an interrupt pulse. We only support modelling edge-triggered interrupts. The value passed is formatted as follows: bits[23:16] the ras error group index (if multiple error groups) (0 atm) bits[15:8] node id bits[7:0] the ras record index in the node.
irq_out_ras_fhi_as_value	master	Value	RAS Fault Handling Interrupt value port. The number of RAS error record groups is dependent on the configuration. As each error record group could have its own interrupts then we write to this value port the error record group number (which is dependent on the configuration) to simulate an interrupt pulse. We only support modelling edge-triggered interrupts. The value passed is formatted as follows: bits[23:16] the ras error group index (if multiple error groups) (0 atm) bits[15:8] node id bits[7:0] the ras record index in the node.
irq_out_rl	master	Signal	Pulsed interrupt output signal combined from all realm (non-RAS) interrupts. This delivers a pulse if any of the interrupt pins of the specified world also deliver a pulse. SW will have to poll all the different reasons to see why it was delivered.
irq_out_s	master	Signal	Pulsed interrupt output signal combined from all secure (non-RAS) interrupts. This delivers a pulse if any of the interrupt pins of the specified world also deliver a pulse. SW will have to poll all the different reasons to see why it was delivered.
l0gptsz_s	slave	Value	RME: This is a four bit signal that encodes the region size that a single LOGPT entry covers. The default value of this port is derived from the parameter rme_l0gpt_entry_covers_log2size_in_bytes which is in a different format to the port. If a valid value is driven then it will be put in the field SMMU_ROOT_GPT_BASE_CFG.LOGPTSZ. The port uses the same encoding as the field. If an invalid value is driven to this pin and legacy_tz_en is low then the model will obey the setting of the parameter out_of_range_l0gptsz_s. This port is sampled at negedge of reset_in and must be set before the negedge of the reset signal.
legacy_tz_en	slave	Signal	For an RME-enabled SMMU then tie this high to get non-RME behaviour See also the parameter SMMU_ROOT_IDR0. This port is sampled at negedge of reset_in and must be set before the negedge of the reset signal.
pvbus_control_s	slave	PVBus	Register subordinate port
pvbus_id_routed_m	master	PVBus	This is a special "id-routed" port for transmitting ATC invalidates and PRI Responses upstream into the PCIe EndPoints, it is not a normal bus. The FastSim ATC invalidate protocol specifies how to route and deal with this this port. See the parameter output_id_routed_transform It is assumed that the StreamID can uniquely route the transaction if there are multiple PCIe Root Complexes.
pvbus_m_tw_msi_qs	master	PVBus	Manager port used for Table Walks, MSIs and Queue access when separate_tw_msi_qs_port==true
pvbus_m	master	PVBus	The TBU manager ports that carry transactions that have been translated from the correspondingly numbered pvbus_s[] port.

Port	Direction	Protocol	Description
pvbus_s	slave	PVBus	The TBU subordinate ports that receive transactions to be translated. They will exit the SMMU through the same numbered pvbus_m[] port.
reset_in	slave	Signal	Reset signal
sev_out	master	Signal	Event signal

## Parameters for SMMUv3AEM

### **NoStreamID\_for\_unaware\_of\_RME**

NoStreamID transactions are usually an RME feature.

An RME-aware IP can usually be configured to turn off the RME features. NoStreamID transactions are always allowed in this case and this parameter has no effect.

This parameter configures the behavior when the IP does not support RME in any configuration.

**0**

Produce an error message and make the model unusable. Receiving a NoStreamID in this configuration is considered an error in the way the platform has been constructed.

**1**

Before SMMUv3.4, allow NoStreamID transactions through. From SMMUv3.4 onwards, the behavior is the same as 2.

**2**

Allow NoStreamID transactions through, forcing the result to Privileged-Data on the downstream bus.

Type: unsigned

Default value: 0

### **PRESET\_REL\_base\_address**

If using preset addresses (SMMU\_IDR1.QUEUES\_PRESET/TABLES\_PRESET) then the queue and table base registers become fixed. If SMMU\_IDR1.REL then the addresses are relative to the base of the register file and this parameter tells the model what address to add to the queue/table addresses to calculate the actual address.

This is for 'embedded implementations' where the memory for these structures is held within the SMMU itself or in a 'close' RAM. The model does not contain any RAM and the integrator must supply a RAM at the appropriate address.

If the preset tables/queues overlap, the RAM has to implement separate secure and non-secure address spaces.

See also:

- TABLES\_PRESET\_smmu\_{,s,r}strtab\_base

- `TABLES_PRESET_smmu_{,s,r}strtab_base_cfg`
- `QUEUES_PRESET_smmu_{,s,r}cmdq_base`
- `QUEUES_PRESET_smmu_{,s,r}eventq_base`
- `QUEUES_PRESET_smmu_{,r}priq_base` (no secure PRIQ).

Type: `uint64_t`

Default value: 0

#### **`QUEUES_PRESET_smmu_cmdq_base`**

If `SMMU_IDR1.QUEUES_PRESET == 1`, this is the value that appears in `SMMU_CMDQ_BASE` and `SMMU_CMDQ_BASE` becomes read-only.

See also parameter `PRESET_REL_base_address`.

Type: `uint64_t`

Default value: 0

#### **`QUEUES_PRESET_smmu_eventq_base`**

If `SMMU_IDR1.QUEUES_PRESET == 1`, this is the value that appears in `SMMU_EVENTQ_BASE` and `SMMU_EVENTQ_BASE` becomes read-only.

See also parameter `PRESET_REL_base_address`.

Type: `uint64_t`

Default value: 0

#### **`QUEUES_PRESET_smmu_priq_base`**

If `SMMU_IDR1.QUEUES_PRESET == 1` and `SMMU_IDR0.PRI == 1`, this is the value that appears in `SMMU_PRIQ_BASE` and `SMMU_PRIQ_BASE` becomes read-only.

See also parameter `PRESET_REL_base_address`.

Type: `uint64_t`

Default value: 0

#### **`QUEUES_PRESET_smmu_r_cmdq_base`**

If `SMMU_IDR1.QUEUES_PRESET == 1`, this is the value that appears in `SMMU_R_CMDQ_BASE` and `SMMU_R_CMDQ_BASE` becomes read-only.

See also parameter `PRESET_REL_base_address`.

Type: `uint64_t`

Default value: 0

#### **QUEUES\_PRESET\_smmu\_r\_eventq\_base**

If SMMU\_IDR1.QUEUES\_PRESET == 1 and SMMU\_ROOT\_IDR0.REALM\_IMPL == 1, this is the value that appears in SMMU\_R\_EVENTQ\_BASE and SMMU\_R\_EVENTQ\_BASE becomes read-only.

See also parameter PRESET\_REL\_base\_address.

Type: uint64\_t

Default value: 0

#### **QUEUES\_PRESET\_smmu\_r\_priq\_base**

If SMMU\_IDR1.QUEUES\_PRESET == 1 and SMMU\_IDR0.PRI == 1 and SMMU\_ROOT\_IDR0.REALM\_IMPL == 1, this is the value that appears in SMMU\_PRIQ\_BASE and SMMU\_PRIQ\_BASE becomes read-only.

See also parameter PRESET\_REL\_base\_address.

Type: uint64\_t

Default value: 0

#### **QUEUES\_PRESET\_smmu\_s\_cmdq\_base**

If SMMU\_IDR1.QUEUES\_PRESET == 1 and SMMU\_S\_IDR1.SECURE\_IMPL == 1, this is the value that appears in SMMU\_S\_CMDQ\_BASE and SMMU\_S\_CMDQ\_BASE becomes read-only.

See also parameter PRESET\_REL\_base\_address.

Type: uint64\_t

Default value: 0

#### **QUEUES\_PRESET\_smmu\_s\_eventq\_base**

If SMMU\_IDR1.QUEUES\_PRESET == 1 and SMMU\_S\_IDR1.SECURE\_IMPL == 1, this is the value that appears in SMMU\_S\_EVENTQ\_BASE and SMMU\_S\_EVENTQ\_BASE becomes read-only.

See also parameter PRESET\_REL\_base\_address.

Type: uint64\_t

Default value: 0

#### **SMMU\_AIDR**

SMMU\_AIDR contains the Major and Minor architectural revision numbers.

Type: uint32\_t

Default value: 0

### **SMMU\_IDR0**

SMMU\_IDR0. The following fields are further combined with the port `conf_system_supports_{sev,httu,btm,cohacc}`:

- `sev`
- `httu`
- `btm`
- `cohacc`



SMMU\_IDR0.RME\_IMPL is the value that the SMMU should have if it is currently RME-aware. It is forced to zero if the SMMU has been forced to be unaware of RME by `legacy_tz_en`.

---

Type: `uint32_t`

Default value: N/A

### **SMMU\_IDR1**

SMMU\_IDR1.

Type: `uint32_t`

Default value: N/A

### **SMMU\_IDR2**

SMMU\_IDR2 holds the BA\_VATOS field.

Type: `uint32_t`

Default value: 0

### **SMMU\_IDR3**

SMMU\_IDR3 is reserved.

Type: `uint32_t`

Default value: 0

### **SMMU\_IDR4**

SMMU\_IDR4 is Imp def.

Type: `uint32_t`



Default value: 0

### **SMMU\_IDR5**

SMMU\_IDR5 contains, amongst others, the Output Address encoded Size (OAS).

Type: `uint32_t`

Default value: 0

### **SMMU\_IDR6**

SMMU\_IDR6 is **RES0** if Enhanced Command Queues do not exist (`SMMU_IDR1.ECMDQ == 0`).

Otherwise, SMMU\_IDR6 contains information about the configuration of the ECMDQs.

Type: `uint32_t`

Default value: 0

### **SMMU\_IIDR**

SMMU\_IIDR contains fields for the implementer, product revision, and so on.

Type: `uint32_t`

Default value: 0

### **SMMU\_MPAMIDR**

In SMMUv3.2, if `SMMU_IDR3.MPAM == 1` then SMMU\_MPAMIDR holds further ID information for the Memory Partitioning And Monitoring (MPAM) extension.

This is optional in SMMUv3.2 and is backported to SMMUv3.1.

Type: `uint32_t`

Default value: 0

### **SMMU\_ROOT\_IDR0**

If SMMU\_ROOT\_IDR0 is 0 then the SMMU is RME-unaware, otherwise the following information applies.

`legacy_tz_en` is a pin that when high disables RME and the SMMU\_ROOT\_IDR0 register reads as zero.

The effective value of `legacy_tz_en` is derived from:

- The last signalled value sampled at negedge of reset
- Or, if never signalled, the inverse of ROOT\_IMPL (bit[0]) of this parameter.

Thus, `ROOT_IMPL` should be zero if we want `legacy_tz_en` to start as high regardless of the actual configuration we want in the `SMMU_ROOT_IDR0` register when the SMMU is RME-aware.

In other words, if the SMMU is to be RME-aware, then all parameters should be configured as though the SMMU is currently RME-aware with the exception that `SMMU_ROOT_IDR0.ROOT_IMPL` is the inverse of the default value of `legacy_tz_en`.

`SMMU_ROOT_IDR0.BGPTM` is the default value of the pin `conf_system_supports_bgptm`.

Type: `uint32_t`

Default value: 0

### **SMMU\_ROOT\_IIDR**

The value of the `SMMU_ROOT_IIDR` register. If it is zero then it is the same as `SMMU_IIDR`.

Type: `uint32_t`

Default value: 0

### **SMMU\_R\_AIDR**

The value of `SMMU_R_AIDR`.

Type: `uint32_t`

Default value: 0

### **SMMU\_R\_IDR0**

The value of `SMMU_R_IDR0`.

Type: `uint32_t`

Default value: 0

### **SMMU\_R\_IDR3**

The value of `SMMU_R_IDR3`.

Type: `uint32_t`

Default value: 0

### **SMMU\_R\_IDR6**

The value of `SMMU_R_IDR6` that configures the ECMDQs.

Type: `uint32_t`

Default value: 0

**SMMU\_R\_MECIDR**

The value of SMMU\_R\_MECIDR.

Type: `uint32_t`

Default value: 0

**SMMU\_R\_MPAMIDR**

This parameter is the value of SMMU\_R\_MPAMIDR, or if ~0u11 then it has the following default values:

- PARTID\_MAX/PMG\_MAX from the SMMU\_MPAMIDR
- HAS\_MPAM\_NS from the SMMU\_S\_MPAMIDR if it exists, otherwise 0.



Note

This parameter is 64 bits but the ID register is 32 bits.

---

Type: `uint64_t`

Default value: 0xFFFFFFFFFFFFFFFF

**SMMU\_S\_IDR0**

Secure IDR0 register.

Type: `uint32_t`

Default value: 0

**SMMU\_S\_IDR1**

SMMU\_S\_IDR1 Indicates if there is a secure side by bit 31.

Type: `uint32_t`

Default value: N/A

**SMMU\_S\_IDR2**

SMMU\_S\_IDR2 Reserved.

Type: `uint32_t`

Default value: 0

**SMMU\_S\_IDR3**

SMMU\_S\_IDR3 Reserved.

Type: `uint32_t`

Default value: 0

### **SMMU\_S\_IDR4**

SMMU\_S\_IDR4 IMP DEF.

Type: `uint32_t`

Default value: 0

### **SMMU\_S\_IDR6**

SMMU\_S\_IDR6 is **RES0** if Secure Enhanced Command Queues do not exist (SMMU\_S\_IDR0.ECMDQ == 0).

Otherwise, SMMU\_S\_IDR6 contains information about the configuration of the ECMDQs.

Type: `uint32_t`

Default value: 0

### **SMMU\_S\_MPAMIDR**

In SMMUv3.2, if SMMU\_IDR3.MPAM == 1 then SMMU\_S\_MPAMIDR holds further ID information for the Memory Partitioning And Monitoring (MPAM) extension.

This is optional in SMMUv3.2 and is backported to SMMUv3.1.

Type: `uint32_t`

Default value: 0

### **TABLES\_PRESET\_smmu\_r\_strtab\_base**

If SMMU\_IDR1.TABLES\_PRESET == 1 and SMMU\_ROOT\_IDR0.REALM\_IMPL == 1, this is the value that appears in SMMU\_R\_STRTAB\_BASE and SMMU\_R\_STRTAB\_BASE becomes read-only.

See also parameter `PRESET_REL_base_address`.

Type: `uint64_t`

Default value: 0

### **TABLES\_PRESET\_smmu\_r\_strtab\_base\_cfg**

If SMMU\_IDR1.TABLES\_PRESET == 1 and SMMU\_ROOT\_IDR0.REALM\_IMPL == 1, this is the value that appears in SMMU\_R\_STRTAB\_BASE\_CFG and SMMU\_R\_STRTAB\_BASE\_CFG becomes read-only.

See also parameter `PRESET_REL_base_address`.

Type: `uint32_t`

Default value: 0

#### **TABLES\_PRESET\_smmu\_s\_strtab\_base**

If `SMMU_IDR1.TABLES_PRESET == 1` and `SMMU_S_IDR1.SECURE_IMPL == 1`, this is the value that appears in `SMMU_S_STRTAB_BASE` and `SMMU_S_STRTAB_BASE` becomes read-only.

See also parameter `PRESET_REL_base_address`.

Type: `uint64_t`

Default value: 0

#### **TABLES\_PRESET\_smmu\_s\_strtab\_base\_cfg**

If `SMMU_IDR1.TABLES_PRESET == 1` and `SMMU_S_IDR1.SECURE_IMPL == 1`, this is the value that appears in `SMMU_S_STRTAB_BASE_CFG` and `SMMU_S_STRTAB_BASE_CFG` becomes read-only.

See also parameter `PRESET_REL_base_address`.

Type: `uint32_t`

Default value: 0

#### **TABLES\_PRESET\_smmu\_strtab\_base**

If `SMMU_IDR1.TABLES_PRESET == 1`, this is the value that appears in `SMMU_STRTAB_BASE` and `SMMU_STRTAB_BASE` becomes read-only.

See also parameter `PRESET_REL_base_address`.

Type: `uint64_t`

Default value: 0

#### **TABLES\_PRESET\_smmu\_strtab\_base\_cfg**

If `SMMU_IDR1.TABLES_PRESET == 1`, this is the value that appears in `SMMU_STRTAB_BASE_CFG` and `SMMU_STRTAB_BASE_CFG` becomes read-only.

See also parameter `PRESET_REL_base_address`.

Type: `uint32_t`

Default value: 0

#### **all\_error\_messages\_through\_trace**

Some conditions in the SMMU are so strange that the software programming the SMMU has done something wrong. At this point messages are output to either `ArchMsg.Error.*` or

`ArchMsg.Warning.*` or to the error stream of the simulator. Outputting to the error stream of the simulator may cause it to return with a non-zero exit status.

If you set this option to true then instead of using the error stream of the simulator it will always use a trace stream allowing the simulation to exit with a zero exit status.

Type: `bool`

Default value: `false`

### **`allow_non_secure_access_to_SMMU_S_INIT`**

If the system has no software operating as a secure agent, set this parameter. This allows non-secure accesses to the `SMMU_S_INIT` register and allows the non-secure software to reset the TLB, clearing out any 'secure' TLB entries.

If the SMMU does not implement the security extensions (`SMMU_S_IDR1.SECURE_IMPL == 0`) then this parameter is ignored.

Type: `bool`

Default value: `false`

### **`apply_ste_instcfg_privcfg_on_all_ats_translated_accesses`**

This parameter is ignored if either of the following are true:

- `SMMU_IDR1.ATTR_PERMS_OVR == 0`
- `SMMU_IDR3.PASIDTT == 1` and the transaction has a PASID

Otherwise, if this parameter is:

#### **`false`**

`STE.INSTCFG/PRIVCFG` is only applied to `ATS-TranslatedTransactions` if `STE.EATS==split-stage`.

#### **`true`**

`STE.INSTCFG/PRIVCFG` is applied to all `ATS-TranslatedTransactions` regardless of the value of `STE.EATS`.

Type: `bool`

Default value: `false`

### **`ats_split_stage_dbm_update_do_with_ATSRequest`**

When doing split-stage ATS, the DBM update for the final stage 2 descriptor can be done either whilst processing the ATS request or delayed until it actually sees the PCIe Translated Transaction using the stage 2 descriptor.

#### **`0`**

Do it when the actual transaction is seen

**1**

Do it when processing the ATS request

**2**

Do it randomly with 50% chance.

Type: unsigned

Default value: 0

**axi\_stream\_msi\_TDEST**

ID of the AXI stream subordinate port on the GIC that receives SMMU originated MSIs sent directly. The name of the GIC port is `axi_stream_msi_s`.

**Note**

If `axi_stream_msi_addr_to_match[1:0] != 0`, or the address is above OAS then this is effectively disabled.

See also:

- `axi_stream_msi_addr_to_match`
- `axi_stream_msi_TID`.

Type: `uint32_t`

Default value: 0

**axi\_stream\_msi\_TID**

ID of the AXI stream manager port that sends SMMU TCU originated MSIs directly to the GIC. The name of the SMMU port is `axi_stream_msi_m`.

**Note**

If `axi_stream_msi_addr_to_match[1:0] != 0`, or the address is above OAS then this is effectively disabled.

See also:

- `axi_stream_msi_addr_to_match`
- `axi_stream_msi_TDEST`.

Type: `uint32_t`

Default value: 0

### **axi\_stream\_msi\_addr\_to\_match**

If the last two bits are 0, any SMMU-originated MSI which exactly matches the address will be sent through the AXI stream port `axi_stream_msi_m` which is usually connected to the GIC.

This parameter drives the value of the `axi_stream_msi_addr_to_match_s` port at simulation reset. For every reset after that, the value of the port will be sampled and used if changed.



The entire address must match, including bits [1:0]. As MSIs are 32 bit aligned then if `axi_stream_msi_addr_to_match[1:0] != 0`, or the address is above OAS then this is effectively disabled.

---

See also:

- `axi_stream_msi_TID`
- `axi_stream_msi_TDEST`.

Type: `uint64_t`

Default value: `0xFFFFFFFFFFFFFFFF`

### **behaviour\_of\_sampled\_at\_reset\_signals**

Some configuration signals into the SMMU are sampled on negedge of reset.

However, it can sometimes be hard to arrange to drive a configuration pin before the negedge of reset.

The configuration pins are sampled:

**0**

at negedge reset.

**1**

at negedge reset, but if a later change occurs at the same simulated time, and no transactions have occurred, then they will be resampled and the SMMU reset again.

Type: `unsigned`

Default value: 0

### **cmdq\_max\_number\_of\_commands\_to\_buffer**

The command queues can buffer fetched commands before issuing them. This parameter is roughly the maximum number of commands to do this for. The programmer visible effects are that just because the CONS pointer shows a command has been consumed does not necessarily mean that it has been issued (and completed). Higher values accentuate this effect.

Type: `uint32_t`



Default value: 10

### **dh\_cmo\_behavior**

DestructiveHint (DH) is a Cache Maintenance Operation (CMO) but some aspects of it differ from normal CMOs. Normal CMOs do not have an incoming memory type or an outgoing memory type.

It can have one of the following values:

**1**

DH is a pure CMO. On the bus, the output memory type is the input memory type.

**2**

DH has both an input and output memory type. The output memory type is calculated as for a normal read.

See also `imp_def_dh_memory_types_supported`.

Type: unsigned

Default value: 1

### **dpt\_configure\_ATS\_formed\_entries**

A comma-separated list of options.

If the string is "" or "none", then no ATS-formed DPT TLB entries are used.

Otherwise the following information applies.

One of:

**"all\_enabled\_stages" or ""**

Use all enabled VMSA stages (default).

**"last\_enabled\_stage"**

Use only the last enabled VMSA stage.

The DPT entry size, one of:

**"vmsa\_and\_gpc" or ""**

Store as the smaller of the VMSA and GPC region (default).

**"vmsa\_only"**

Store as the VMSA region.

ATS-formed entries and DPT-formed entries can be distinguishable, one of:

**"on\_fault\_always\_walk" or ""**

ATS-formed and DPT-formed entries are not distinguished (default).

**"on\_fault\_walk\_if\_ATS\_formed"**

ATS-formed entries are not definitive and cause a walk. DPT-formed entries are definitive and do not cause a walk.

An ATS-formed entry can still be inserted if the only reason it fails is because of the final GPC check, one of:

- “only\_if\_passes\_final\_PA\_GPC\_check”
- “even\_if\_fails\_final\_PA\_GPC\_check”

For example:

“all\_enabled\_stages, vmsa\_and\_gpc”

If we find a DPT TLB entry that is ATS-formed or the implementation does not distinguish between ATS-formed and DPT-formed entries then:

- For downgradeable transactions, if the entry does not allow the transaction but would allow the downgraded transaction:

**“do\_not\_prefer\_downgrade\_over\_DPT\_walk”**

Walk the DPT to see if the non-downgraded transaction would be allowed.

**“prefer\_downgrade\_over\_DPT\_walk” (default)**

Just do the downgrade and avoid a DPT walk.

- For a NOPpable transaction, if the entry does not allow the transaction:

**“do\_not\_prefer\_found\_entry\_NOP\_over\_DPT\_walk”**

Walk the DPT to see if the original transaction is allowed.

**“prefer\_found\_entry\_NOP\_over\_DPT\_walk” (default)**

Just **NOP** the transaction and avoid a DPT walk.



Note

If no entry is found, a NOPpable transaction still performs a walk.

---

Type: string

Default value: “all\_enabled\_stages, vmsa\_and\_gpc”

### **dpt\_configure\_invalidation**

Configure when entries are invalidated when performing a DPT check.

Specify a comma-separated list of options. Choose one of:

- lookup\_fault\_invalidates\_any\_existing\_entries or “” (default)
- lookup\_fault\_leaves\_any\_existing\_entries

Choose one of:

- noaccess\_fault\_invalidates\_any\_existing\_entries or “” (default)
- noaccess\_fault\_leaves\_any\_existing\_entries.

Type: `string`

Default value: `""`

### **`dr_downgrade_behavior`**

There are certain circumstances under which a DR (DestructiveRead) can downgrade to either an ordinary read or a RCI (read-with-clean-and-invalidate).

In cases where there is a choice, the following values apply:

**0**

Always downgrade to ordinary read

**1**

Downgrade to RCI

**2**

Randomly choose on a per-transaction basis.

Type: `unsigned`

Default value: `0`

### **`enable_device_id_checks`**

If this parameter is true then the DeviceIDs seen by the GIC are:

- **for client devices**

`DeviceID = StreamID + translated_device_id_base`

- **for SMMU-generated MSIs**

`smmu_msi_device_id`

This parameter enables two checks:

- If the DeviceID is used in the `output_attribute_transform` parameter, if it overflows 32 bits then the model warns. If the DeviceID is not used, it is assumed that the external agent that forms the DeviceID warns if it overflows.
- If the SMMU supports MSIs, the model checks that the GIC is able to distinguish an MSI generated by the SMMU from one generated by a client device.

As the exact mechanism to determine the DeviceID is in the system and not necessarily under control of the SMMU then you can disable these warnings using this parameter.

See also the parameters: `output_attribute_transform` and `msi_attribute_transform`.

Type: `bool`

Default value: `true`

**hide\_warning\_ACCESSEN\_GPCEN\_set\_to\_1\_in\_a\_single\_write**

The architecture recommends against setting `SMMU_ROOT_CR0.ACCESSEN` and `SMMU_ROOT_CR0.GPCEN` to 1 in the same write if it is possible that a concurrent client device transaction could appear at the SMMU. This is because there could be a time window where the client device transaction sees effective values `ACCESSEN == 1` and `GPCEN == 0` and so would bypass GPC checking.

If your system cannot generate this possibility then you can set this parameter to true and turn off the warning that software has set both `ACCESSEN` and `GPCEN` to 1 at the same time.

Type: `bool`

Default value: `false`

**hide\_warning\_EOPD\_differs\_from\_what\_would\_be\_cached**

When this parameter is set to true, warnings that the effective EOPD value differs from what would be cached in the TLB are disabled. False (warnings are showed) by default.

Type: `bool`

Default value: `false`

**hide\_warning\_NoStreamID\_transaction\_for\_unsupported\_PAS\_or\_MPAM\_SP**

When RME is not supported then a NoStreamID transaction with `PAS[1] == 1` OR `MPAM_SP[1] == 1` is treated as though `PAS[1] == 0` and `MPAM_SP[1] == 0`. This is usually a system construction error and is not expected to occur.

The SMMU warns when this occurs, but the warning can be hidden by setting this parameter.

Type: `bool`

Default value: `false`

**howto\_identify**

If `use-identify` then the SMMU uses the `identify` port to determine the `ssd`, `StreamID`, `subStreamID`. Otherwise, this string extracts them from the transaction's attributes.

Examples:

```
SEC_SID=ExtendedID[63], SSV=ExtendedID[62], SubstreamID=ExtendedID[51:32],
StreamID=ExtendedID[31:0]
```

or

```
nSEC_SID=ExtendedID[63], StreamID=ExtendedID[55:24], nSSV=ExtendedID[20],
SubstreamID=ExtendedID[19:0]
StreamID[31:24]=0, StreamID[23:0]=ExtendedID[23:0], SSV=1[0], ...
```



If you are not using the `use-identify` option then the configuration string is parsed according to the rules in the section ‘Parameters for parsing transaction attributes’.

LHS Symbols:

**SIDV**

Indicates that the StreamID is valid.

**SSV**

Indicates that the SubstreamID is valid.

**SEC\_SID / SEC\_SID\_bit\_1**

Bits 0 and 1 respectively of the StreamID Security State

**StreamID**

(32 b) valid if `SIDV` is 1 or both `SIDV` and `nSIDV` are unused.

**SubstreamID**

(20 b) is valid if `ssv` is true.

**nSEC\_SID / nSEC\_SID\_bit\_1 / nSSV / nSIDV**

Negative logic of above parameters. Different attributes are independent and can use negative or positive logic.

RHS Symbols:

**ExtendedID / ManagerID64 / UserFlags**

Incoming PVBUS transaction attributes

`SEC_SID` is 1b in the model. For RME systems, in hardware, then `SEC_SID` is 2b, in the model `SEC_SID_bit_1` represents bit[1].

`SIDV == 0` indicates a NoStreamID transaction and the SSD is the PAS of the transaction, `SEC_SID` is not used.

Negative and positive logic symbols for the same attribute is an error.

The model uses the term SSD (Security State Determination) to mean which security state the transaction, register, structure, event, etc. belongs to.

In the SMMUV3 Architecture, the side-band signal `SEC_SID` holds the information for the transactions, but uses a different encoding.

Before RME, `SEC_SID` was a one bit signal. With RME, in the hardware, it was expanded to 2b. To retain backwards compatibility in the model, `SEC_SID` remains as 1b in this parameter, but the second bit can be expressed with `SEC_SID_bit_1` (and its negative logic version `nSEC_SID_bit_1`)

Security state	SSD	SEC_SID_bit_1	SEC_SID
secure	0	0	1

Security state	SSD	SEC_SID_bit_1	SEC_SID
non-secure	1	0	0
root (reserved)	2	1	1
realm	3	1	0

For those systems that support NoStreamID transactions, and `howto_identify` is not using the port, first the SMMU determines the value of `SIDV` or `NSIDV` to see if the transaction is a NoStreamID transaction (`SIDV == 0` or `NSIDV == 1`). If so then the rest of the `howto_identify` string is ignored as the information extracted relates only to StreamID transactions. Instead more information is extracted using the parameter `howto_identify_NoStreamID_extra_info`.

In AMBA systems, the equivalent signal to `SIDV` is called either `ARMMUVALID` or `AWMMUVALID`. Collectively they are known as `AxMMUVALID`.

Type: string

Default value: "use-identify"

### **`howto_identify_NoStreamID_extra_info`**

The behavior of this parameter depends on `howto_identify`

- if it equals 'use-identify' then this must be "", otherwise there is an error.
- if it identifies a NoStreamID transaction (`SIDV=0`) then this parameter includes one or more of
  - `MPAM_SP`
  - `MPAM_PARTID`
  - `MPAM_PMG`
  - `MECID`
  - `HWATTR_KIND_0`
- in any other case, this parameter is ignored.

Fields set in this parameter must not overlap the `SIDV/NSIDV` fields in `howto_identify`

Example:

```
MPAM_PMG[7:0]=ExtendedID[62:55], MPAM_PARTID[15:0]=ExtendedID[54:39],
MPAM_SP[1:0]=ExtendedID[38:37], MECID[15:0]=UserFlags[31:16]
HWATTR_KIND_0[3:0]=ExtendedID[42:39]
```



The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

Type: string

Default value: ""

### **httu\_early\_st2\_permission\_fault\_if\_af\_update\_at\_stagel**

If a stage 1 descriptor needs an HTTU update, but the descriptor is unwriteable at stage 2 and also a stage 1 permission fault occurs, then the architecture permits either the stage 1 or stage 2 permission fault to be recorded.

**0**

Stage 1 permission check. Check stage 1 descriptor writeable at stage 2 if AF- or DBM-update required.

**1**

Check stage 1 descriptor writeable at stage 2 if AF-update required. Stage 1 permission check. Check stage 1 descriptor writeable at stage 2 if DBM-update required.

**2**

Do behavior 1 or 2 randomly with a 50% chance.

Type: unsigned

Default value: 0

### **httu\_memory\_types\_supported**

This is a comma-separated list of memory types that are **IMPLEMENTATION DEFINED** as supporting HTTU. However, the system must have Far Atomic support for the specified memory address and memory type.

Device types:

- nGnRnE
- nGnRE
- nGRE
- GRE

Normal memory types are composed of an 'inner' and an 'outer' cacheability. The model only supports types where the inner and outer are identical.

- Normal non-cacheable types are nc\_nb, nc
- Cacheable types are of the form (na?|(ra)?(wa)?)(WT|WB)(tr)?:

**na/ra/wa**

no/read/write allocate

**WT/WB**

Write through/write back

**tr**

Transient

Exceptions:

- na and tr are incompatible
- Without na, you must specify at least one of ra or wa. For example, “WT” is illegal, “raWT” is legal.

rawaWB is always supported and is optional.

Examples:

**“rawaWB, raWB, waWB, naWB”**

Only the WB type is supported.

**“nc”**

rawaWB and the normal non-cacheable type are supported.

Type: `string`

Default value: `"rawaWB, raWB, waWB, naWB"`

### **`imp_def_L1CD_L2Ptr_out_of_range`**

If an L1CD.L2Ptr is out of range of IAS/OAS as appropriate then what happens is controlled by this parameter:

**0**

If it is an IPA, then Stage 2 Translation Fault. If it is a PA then truncate to OAS.

**1**

Generate C\_BAD\_SUBSTREAMID if an IPA and > IAS, or if a PA and > OAS.

**2**

Generate C\_BAD\_SUBSTREAMID if an IPA and > IAS, or F\_CD\_FETCH if a PA and > OAS.

**3**

Truncate the IPA or PA to IAS/OAS as appropriate.



If the model is configured as SMMUv3.1 then this parameter is IGNORED, and the model behaves as though this parameter was set to 1. The SMMUv3.1 architecture allows more behaviors but the model only implements this one.



SMMUv3.0 allows more behaviors than can be expressed by this parameter.

Type: `unsigned`

Default value: 0



**imp\_def\_MTCOMB\_ATSResponse\_N\_bit**

If `SMMU_IDR3.MTCOMB == 0`, the SMMU architecture always sets the N field in ATS Responses to 0. When `SMMU_IDR3.MTCOMB == 1`, Arm recommends the N field is set to 1 when either of the following applies:

- The ATS Translation Request is Forced-WB.
- The ATS Translation Request is subject to stage 1 and both of the following apply:
  - `CD.MTOP` is configured to replace the incoming memory type (0).
  - The resultant memory type is Normal-iWB-oWB

This parameter controls how the model obeys the recommendation:

**0**

The recommendation is followed.

**1**

Only the first bullet point is followed.

**2**

None of the recommendation is followed.

Type: unsigned

Default value: 0

**imp\_def\_PID0**

If `imp_def_has_PID_CID` is true then this is the PID0 value.

Type: unsigned

Default value: 0x83

**imp\_def\_PID1**

If `imp_def_has_PID_CID` is true then this is the PID1 value.

Type: unsigned

Default value: 0xb4

**imp\_def\_PID2**

If `imp_def_has_PID_CID` is true then this is the PID2 value.

Type: unsigned

Default value: 0xb

**imp\_def\_PID3**

If `imp_def_has_PID_CID` is true then this is the PID3 value.

Type: unsigned

Default value: 0x0

### **imp\_def\_PID4**

If `imp_def_has_PID_CID` is true then this is the PID4 value.

Type: unsigned

Default value: 0x4

### **imp\_def\_S1ContextPtr\_out\_of\_range**

If an STE is fetched that uses a stage 1 then if either:

- Stage 1 only and `S1ContextPtr > OAS`, or
- Stage 1+2 and `S1ContextPtr > IAS`

what happens is IMP DEF and this parameter controls the behavior:

**0**

Stage 1 only: `C_BAD_STE`. Stage 1+2: `C_BAD_STE`.

**1**

Stage 1 only: `C_BAD_STE`. Stage 1+2: truncate to IAS.

**2**

Stage 1 only: truncate to OAS. Stage 1+2: `C_BAD_STE`.

**3**

Stage 1 only: truncate to OAS. Stage 1+2: truncate to IAS.

**4**

Stage 1 only: truncate to OAS. Stage 1+2: stage 2 translation fault.

**5**

Stage 1 only: `C_BAD_STE`. Stage 1+2: stage 2 translation fault.

The architecture also allows for `F_CD_FETCH` but the model does not support this.



**Note**

In SMMUv3.1, the only allowed values of this parameter are 0 or 5.

---

Type: unsigned

Default value: 0

**imp\_def\_alloccfg**

STE.ALLOCCFG overrides the read/write/transient hints on cacheable types. However these are hints and an implementation may choose to treat them differently.

**0**

Apply the alloc hints as architecturally specified.

**1**

Ignore all ALLOCCFG fields (treated as zero).

**2**

Apply ALLOCCFG only when MTCFG == 1.

Type: unsigned

Default value: 0

**imp\_def\_apply\_dre\_dcp\_to\_full\_ats**

STE.DRE and STE.DCP control the downgrade of certain transactions to **NOP**.

If this parameter is true, for ATS-TranslatedTransactions using STE.EATS == all\_stages or use\_dpt, the STE.DRE/STE.DCP controls are applied.

For ATS-TranslatedTransactions using STE.EATS == eats\_use\_dpt, the controls are always applied.

For SMMUv3.4 and later, or if SMMU\_IDR3.DPT == 1, this parameter is ignored and treated as 'true'.

Type: bool

Default value: false

**imp\_def\_armv82\_64KiB\_granule\_ttd\_bits\_15\_12\_interpretation**

From Armv8.2, 64 KiB granules can support 52b of output PA by using bits[15:12] of the Translation Table Descriptor as bits[51:48] of the output address.

**0**

These bits are always used independently of OAS or PS.

**1**

These bits are ignored if OAS <= 48b.

Type: unsigned

Default value: 0

**imp\_def\_ats\_attribute\_stashing**

The SMMU architecture allows an ATS request to return the attributes with which to make the Translated Access. PCIe does not define any transaction attributes in the Arm sense and so the

mechanism for doing this is IMP DEF. Usually this would be done by packing them into the high order address bits of the return response.

In the model, the representation of the ATS reply returns the attributes directly and it is up to the ATC whether it wants to use them or not.

The parameter configures what to place in those architectural attributes in the ATS Reply:

- 0** The architectural attributes
- 1** Inner Write Back, Outer Write Back, Inner Shared, read and write allocate, User-Data
- 2** Inner Write Back, Outer Write Back, Outer Shared, read and write allocate, User-Data

The SMMU cannot force an ATC to use these attributes. They are simply the attributes that are returned in the non-PCle part of the ATS reply.

Type: unsigned

Default value: 0

#### **imp\_def\_ats\_response\_stu**

A successful ATS Response with RW != 0 can return any-sized region from the STU to the actual region size. The chosen size is:

- 0** Use the full size of the region.
- 1** Use the STU.
- 2** Use half the size of the region.

Type: unsigned

Default value: 0

#### **imp\_def\_cohacc\_effect**

SMMU\_IDR0.COHAACC is a system property. However, the exact nature of the transactions that the SMMU emits is an IMP DEF property when COHAACC == 0:

- 0** COHAACC == 0 forces the output attributes of SMMU-generated accesses to non-shared.
- 1** The only effect of COHAACC is what is reported in SMMU\_IDR0.COHAACC and has no effect on the output attributes of SMMU-generated accesses.

Type: unsigned

Default value: 0

### **imp\_def\_contiguous\_bit\_handling**

If the Contiguous bit is set in a translation table descriptor then modify how it is cached:

**0**

Use the full size of the contiguous region.

**1**

Ignore the contig bit for determining the region size.

**2**

Use half the size of the contiguous region.

See `imp_def_gpt_contiguous_bit_handling` for GPT Contig bit handling.

Type: unsigned

Default value: 0

### **imp\_def\_dh\_memory\_types\_supported**

Destructive Hint (DH) might not be supported for all output memory types.

Which memory types are supported is expressed as either `any` or a comma-separated list of the symbols:

**iWB-oWB**

inner Write-Back, outer Write-Back, any shareability

**iWB-oWB-sh**

inner Write-Back, outer Write-Back, inner or outer shareable only

**iNC-oWB**

inner Normal Non-Cacheable, outer Write-Back, any shareability

**iNC-oWB-sh**

inner Normal Non-Cacheable, outer Write-Back, inner or outer shareable only

**iNC-oNC**

inner Normal Non-Cacheable, outer Non-Cacheable

...

Other variants of Normal memory types

**GRE**

Device-GRE

**nGRE**

Device-nGRE

**nGnRE**

Device-nGnRE

**nGnRnE**

Device-nGnRnE

An empty string is interpreted as `any`.

See also `dh_cmo_behavior`.

Type: `string`

Default value: `"iWB-oWB-sh, iNC-oWB-sh"`

**imp\_def\_effective\_ATTR\_TYPES\_OVR\_is\_false\_per\_port**

`SMMU_IDR1.ATTR_TYPES_OVR == 1` means that the STE and `SMMU_(S_)GBPA MTCFG/SHCFG/ALLOCCFG` have an effect.

However, an implementation is allowed to ignore this being one for specific ports and *not* apply the overrides `MTCFG/SHCFG/ALLOCCFG` despite `SMMU_IDR1.ATTR_TYPES_OVR == 1`.

This parameter is a comma-separated list of port ranges (indexed from 0) for those ports where `SMMU_IDR1.ATTR_TYPES_OVR` behaves as 0. For example:

```
0, 10-20, 40
```

Type: `string`

Default value: `""`

**imp\_def\_gpt\_contiguous\_bit\_handling**

If we find a GPT Contiguous descriptor then modify how it is cached:

**0**

Use the full size of the contiguous region.

**1**

Ignore the `contig` bit for determining the region size and use PGS.

**2**

Use half the size of the contiguous region.

See `imp_def_contiguous_bit_handling` for VMSA `Contig` bit handling.

Type: `unsigned`

Default value: 0

**imp\_def\_has\_PID\_CID**

If this is true then the SMMU model has the standard PID/CID ID registers. Only the PID0..PID4 registers can be customized and the parameters `imp_def_PID0..imp_def_PID4` are used.

Type: `bool`

Default value: `true`

**imp\_def\_no\_InD\_PnU\_on\_downstream\_system**

SMMUv3.4 deprecated the SMMU emitting the Instruction/Data (InD) and the Privileged/User (PnU) markings onto the downstream interconnect because:

- Many interconnects have no way to represent them.
- The downstream system should not care about the value of these fields.

If this parameter is set to true, all transactions are marked as Privileged-Data on the downstream interconnect.

From SMMUv3.4, this parameter is ignored and has the effective value 'true'.

Type: `bool`

Default value: `false`

**imp\_def\_ns\_bit\_for\_s\_gatos\_on\_s1\_bypass\_non\_sel2**

This parameter only has an effect if `SEL2 == 0`.

When `SEL2 == 0`, Secure virtualisation is not supported and only stage 1 is supported for Secure Streams.

In this case, the Secure ATOS interface does not provide a mechanism to specify the input NS bit to the stage 1 translation. The input bit is IMP DEF and only has an effect if the transaction has no SubstreamID and bypasses by S1DSS.

This parameter specifies the IMP DEF input bit:

- 0**  
Secure
- 1**  
Non-secure
- 2**  
Random

See also `imp_def_ns_bit_for_s_vatos_on_s1_bypass` which configures something similar for the Secure VATOS (not GATOS) interface.

Type: `uint32_t`

Default value: 0

### **imp\_def\_ns\_bit\_for\_s\_vatos\_on\_s1\_bypass**

When a Secure VATOS operation for a translation bypasses stage 1 by S1DSS then the output NS bit is the same as the input NS bit of the translation.

The architecture does not provide an input NS bit in the SMMU\_S\_VATOS\_ADDR register and it is treated as an IMP DEF value.

This parameter specifies that value:

**0**

Secure

**1**

Non-secure

**2**

Random

See also `imp_def_ns_bit_for_s_gatos_on_s1_bypass_non_sel2` which configures something similar for the Secure GATOS (not VATOS) interface.

Type: `uint32_t`

Default value: 0

### **imp\_def\_ras\_access\_control\_policy**

The access control of the RAS nodes:

**"" or "use-imp\_def\_ras\_allow\_non\_secure\_accesses\_if\_supports\_secure"**

The parameter `imp_def_ras_allow_non_secure_accesses_if_supports_secure` is used to determine access.

**"non-secure"**

Allow access to all PASes.

**"secure"**

Only allow access to secure and root. If secure is not implemented then allow access to non-secure.

**"root-only"**

The register file is only accessible to root-pas transactions. If root is not implemented then it acts as though this parameter was "use-imp\_def\_ras\_allow\_non\_secure\_accesses\_if\_supports\_secure".

Type: `string`

Default value: `"use-imp_def_ras_allow_non_secure_accesses_if_supports_secure"`



**imp\_def\_ras\_allow\_non\_secure\_accesses\_if\_supports\_secure**

If two security worlds are supported, in other words `SMMU_S_IDR1.SECURE_IMPL == 1` then if this parameter is true, non-secure accesses are allowed to access any RAS registers, see parameter `ras`. Otherwise, non-secure accesses are **RAZ/WI**.

If only a single security state (non-secure) is supported, then this parameter is ignored and non-secure accesses are always allowed.

See also `imp_def_ras_access_control_policy` which can override this parameter.

Type: `bool`

Default value: `false`

**imp\_def\_reset\_unknown\_fields\_to\_zero**

Many fields and registers in the SMMUv3 architecture reset to an **UNKNOWN** value. However, many implementations choose to reset to 0. By setting this parameter to true then those fields are initialised to zero.

Type: `bool`

Default value: `false`

**imp\_def\_rme\_gpf\_syndrome\_for\_PMCG\_MSIs**

An MSI access from a PMCG that experiences a GPF is permitted to be reported as either of:

- REASON = GERROR and FAULTCODE = OTHER\_GPF
- REASON = TRANSACTION

The values of this string are one of:

- `other_gpf`
- `transaction`

See also the parameter `imp_def_rme_gpf_syndrome_for_RAS_MSIs`.

Type: `string`

Default value: `"other_gpf"`

**imp\_def\_rme\_gpf\_syndrome\_for\_RAS\_MSIs**

An MSI access from a RAS record interrupt that experiences a GPF is permitted to be reported as either of:

- REASON = GERROR and FAULTCODE = OTHER\_GPF
- REASON = TRANSACTION

The values of this string are one of:

- other\_gpf
- transaction

See also the parameter `imp_def_rme_gpf_syndrome_for_PMCG_MSIs`.

Type: string

Default value: "other\_gpf"

### **`imp_def_rme_mpam_info_from_NoStreamID_on_gpt_walks`**

The MPAM related fields in `howto_identify_NoStreamID_extra_info` can be used on the GPT walks for NoStreamID transactions.

If empty then `imp_def_rme_mpam_info_from_NoStreamID_on_gpt_walks_ignored` is obeyed.

If non-empty then is a comma-separated list.

One of:

#### **`mpam_sp:pas`**

The PAS is used as the MPAM\_SP.

#### **`mpam_sp:incoming Or ""`**

The incoming MPAM\_SP is used, or 0 if it does not have one.

One of:

#### **`partid/pmg:0`**

The PARTID/PMG is 0

#### **`partid/pmg:incoming`**

The PARTID/PMG is the untruncated incoming, or 0 if it does not have one.

#### **`partid/pmg:truncate_or_0 Or ""`**

The same as `partid/pmg:incoming` but truncated to the appropriate `SMMU_MPAMIDR` / `SMMU_S_MPAMIDR` OR `SMMU_R_MPAMIDR`. If the register does not exist, use 0.

#### **`partid/pmg:truncate`**

The same as `partid/pmg:truncate_or_0` but if the register does not exist, use the maximum bit width implied by all registers that do exist.

See also: `imp_def_rme_mpam_info_on_NoStreamID`.

Type: string

Default value: ""

### **`imp_def_rme_mpam_info_from_NoStreamID_on_gpt_walks_ignored`**

The MPAM related fields set in `howto_identify_NoStreamID_extra_info` are ignored when this parameter is set. This parameter only makes sense when `howto_identify` equals `use-identify` so in any other case it must be false.

When this parameter is set, the MPAM information for GPT walks is:

- MPAM\_SP = PAS
- MPAM\_PARTID = 0
- MPAM\_PMG = 0

This parameter is ignored if `imp_def_rme_mpam_info_from_NoStreamID_on_gpt_walks` is not empty.

Type: `bool`

Default value: `false`

### **`imp_def_rme_mpam_info_on_NoStreamID`**

The MPAM related fields in `howto_identify_NoStreamID_extra_info` can be used for the MPAM information on downstream NoStreamID transactions.

If non-empty, this is a comma-separated list of options.

One of:

#### **`mpam_sp:pas`**

The PAS is used as the MPAM\_SP.

#### **`mpam_sp:incoming Or ""`**

The incoming MPAM\_SP is used, or 0 if it does not have one.

One of:

#### **`partid/pmg:0`**

The PARTID/PMG is 0

#### **`partid/pmg:incoming`**

The PARTID/PMG is the untruncated incoming, or 0 if it does not have one.

#### **`partid/pmg:truncate_or_0 Or ""`**

The same as `partid/pmg:incoming` but truncated to the appropriate `SMMU_MPAMIDR` / `SMMU_S_MPAMIDR` OR `SMMU_R_MPAMIDR`. If the register does not exist, use 0.

#### **`partid/pmg:truncate`**

The same as `partid/pmg:truncate_or_0` but if the register does not exist, use the maximum bit width implied by all of the registers that do exist.

See also: `imp_def_rme_mpam_info_from_NoStreamID_on_gpt_walks`.

Type: `string`

Default value: `""`

### **`imp_def_split_ATS_attributes_is_stage1`**

If using split stage ATS, it is IMP DEF whether the stage 1 attributes are returned to the ATS request or stage 2.

This only has a meaning if the SMMU can stash attributes in the ATS response.

Type: `bool`

Default value: `false`

#### **`imp_def_truncate_out_of_range_streamids_on_invalidate_commands`**

If this parameter is true, the StreamID fields of the following commands are truncated to (S\_)SIDSIZE:

- `CMD_ATC_INV`
- `CMD_CFGI_STE`
- `CMD_CFGI_STE_RANGE`
- `CMD_CFGI_CD`
- `CMD_CFGI_CD_ALL`

Otherwise, these commands will **NOP**.

Type: `bool`

Default value: `false`

#### **`imp_def_v3_atos_fault`**

For an IPA to PA ATOS translation that encounters a Stage 1 Address Size Fault, the `PAR.REASON` field reports:

- In SMMUv3.1, 'Stage 1' (0)
- In SMMUv3.0, 'Stage 1' (0) or 'Input' (3) depending on the implementation.

This parameter is ignored for SMMUv3.1.

For SMMUv3.0, the values are:

**0**

Report as 'Input' (3)

**1**

Report as 'Stage 1' (0).

Type: `unsigned`

Default value: 0

#### **`ish_is_osh`**

When set, any transaction that would use the architectural inner shareable domain is converted to use the outer shareable domain.



This parameter should match the equivalent `ish_is_osh` from the PE. If an incompatible value of the `ish_is_osh` parameter is configured for the PE and the SMMU, data coherency may be compromised.

Type: `bool`

Default value: `false`

### **`list_of_tbu_ports_for_gpc_only`**

This parameter specifies a list of TBU ranges that are configured in GPC-only mode. Any TBU listed here will bypass address translation and perform only granule protection checks. TBUs not listed will operate with standard translation functionality.

Type: `string`

Default value: `""`

### **`mec_attribute_transform`**



The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

If MEC is supported, this is applied to *all* downstream transactions to transport the MEC information.

- `""` or `"none"` – no transform
- How to alter the output attributes. Example:

```
"UserFlags[31:16]=MECID[15:0]"
```

RHS/LHS Symbols:

- `ExtendedID/ManagerID64/UserFlags`.

RHS Symbols:

- `MECID`

Any bits with no transform are unchanged.



Attribute transforms applied before this:

- for client transactions `output_attribute_transform / output_attribute_transform_for_NoStreamID`.
- for table walks `tw_qs_attribute_transform`.

- for MSIs `msi_attribute_transform`.
- if MPAM is enabled `mpam_attribute_transform`.

---

Type: string

Default value: ""

### **mpam\_attribute\_transform**

---



The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

---

If MPAM is supported, this is applied to *all* downstream transactions to transport the MPAM information.

- "" or "none" – no transform
- How to alter the output attributes. Example:

```
"ExtendedID[62:55]=MPAM_PMG, ExtendedID[54:39]=MPAM_PARTID,
ExtendedID[38]=MPAM_SP[0]"
```

RHS/LHS Symbols:

- ExtendedID/ManagerID64/UserFlags.

RHS Symbols:

- MPAM\_PARTID
- MPAM\_PMG
- MPAM\_NS
- MPAM\_SP
- numeric literals

Any bits with no transform are unchanged.



- attribute transforms applied before this:
  - for client transactions `output_attribute_transform` / `output_attribute_transform_for_NoStreamID`.
  - for table walks `tw_qs_attribute_transform`.
  - for MSIs `msi_attribute_transform`.
- `mec_attribute_transform` is applied after this.

- for translated transactions from client devices then `MPAM_NS = ! SEC_SID`.

Type: string

Default value: "ExtendedID[62:55]=MPAM\_PMG, ExtendedID[54:39]=MPAM\_PARTID, ExtendedID[38]=MPAM\_NS"

### **mpam\_sp\_options**

The width of the MPAM\_SP output side-band information.

The SMMU architecture says that the SMMU is a four-space MPAM component when RME-DA is implemented. However, it can potentially be converted to a two-space MPAM at the edge of the SMMU.

This parameter controls the width of the MPAM\_SP output side-band-information:

**1**

1b, conventionally the side-band is then called `MPAM_NS`. Any 2b MPAM\_SP value generated will have bit[1] forced to zero.

**2**

2b, the side-band is 2b.

The same effect can be achieved by using the parameter `mpam_attribute_transform` to only export a single bit of the MPAM\_SP. However, this option allows a model system to be built with a single static version of `mpam_attribute_transform` and then dynamically switch its behavior more simply.

Type: unsigned

Default value: 2

### **msi\_attribute\_transform**

Transform downstream attributes of MSI transactions.



The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

- "" or "none" – no transform
- How to alter output attributes of SMMU-generated MSIs. Example:

```
"UserFlags[15:0]=smmu_msi_device_id[31:16],
ManagerID64[15:0]=smmu_msi_device_id[15:0],
ExtendedID=0"
```

LHS Symbols:

**ExtendedID / ManagerID64 / UserFlags**

Outgoing PVBUS transaction attributes

RHS Symbols:

**smmu\_msi\_device\_id**

The value stored in the parameter with the same name

**interrupt\_kind**

The selected bit corresponds to the interrupts listed below

**0/1**

EVENTQ s/ns

**2**

PRIQ

**3/4**

CMD\_SYNC s/ns

**5/6**

GERROR s/ns

**7/8**

PMCG s/ns

**9/10/11**

RAS FHI/ERI/CRI

**12/13**

gpf\_far/gpt\_cfg\_far

**14/15/16/17**

Realm EVENTQ/PRIQ/CMDQ/GERROR

**HWATTR\_KIND\_0**

PBHA information

**Numeric Literals**

Any number. Ex: 0x1234

ExtendedID/ManagerID64/UserFlags start with values {0, 0xFFFFFFFF, 0} respectively.

Any bits with no transform are unchanged.

This transform can be used to determine the DeviceID passed to the GIC to distinguish MSIs generated by the SMMU from those generated by client devices.



Note

See also `output_attribute_transform` and `enable_device_id_checks`.





After 11.25 the `interrupt_kind` field was extended to 5 bits. This is strictly a non-backwards compatible change. However, the original 3 bits were insufficient to express all the interrupt kinds that exist.

Type: `string`

Default value: `"ExtendedID[31:0]=smmu_msi_device_id, ManagerID64[31:0]=0xFFFFffff"`

### **`msi_ra_wa_tr`**

A bitmap of the Read Allocate, Write Allocate and Transient hints for MSIs to cacheable memory:

#### **`bit[0]`**

Transient

#### **`bit[1]`**

Write Allocate

#### **`bit[2]`**

Read Allocate

If not Write Allocate then it is forced to Read Allocate as a limitation of AMBA.

Type: `uint32_t`

Default value: 7

### **`non_arch_incoming_stronger_than_iWB_oWB_forces_output_iNC_oNC_or_stronger`**

If not empty, this enables a specific non-architectural behavior on the comma-separated list of port indexes, or ranges. For example:

```
0, 10-20, 40
```

In the normal translation process, the input attributes are usually replaced by the attributes from the page tables or SMMU\_(S\_)GBPA.

The behavior is if incoming attributes are iWB-oWB, use the architectural attributes. Otherwise use the stronger of iNC-oNC-osh and the architectural attributes.

This is useful if the ports represent transactions from the PCIe subsystem and the PCIe devices output:

- iWB-oWB if not No\_Snoop -> output is architectural attributes
- iNC-oNC-osh if No\_Snoop -> output is iNC-oNC-osh or stronger.

Type: `string`

Default value: `""`

**normalize\_input\_normal\_non\_iWB\_oWB\_to\_iNC\_oNC\_osh**

When set, use Inner Non Cacheable, Outer Non Cacheable, Outer Shareable for any upstream transaction that would use any of the following attributes:

- Normal Non-cacheable Bufferable
- Normal Non-cacheable Non-bufferable
- Write-through

**Note**

This parameter should match the equivalent configuration from the PE. If an incompatible value of this parameter is configured for the PE and the SMMU, data coherency may be compromised.

Type: `bool`

Default value: `false`

**number\_of\_ports**

The number of port pairs that the SMMU has.

Type: `unsigned`

Default value: `1`

**nw\_dcp\_extra\_drop\_conditions**

NW-DCP is a hint and can be dropped for any reason.

This is a comma-separated list of:

**“INST”**

NW-DCP is architecturally 'data' but by enabling this option then STE.INSTCFG is applied and it can fault due to SIF and, for rl-ssd, preventing instruction access to ns-PAS.

**“SIF-cached”**

NW-DCP needs any of rwx permissions to go downstream. If SIF has been cached into the TLB entry then it will have removed execute permission and so for an execute-only page the NW-DCP would be denied.

If you set this option, for NW-DCP, the SMMU behaves as if SIF was cached.

The parameter `ordering_of_PAN_and_xn_by_ns_pas` can force TLB-caching of SIF and takes precedence over this option.

**“combined-st2”**

When considering the stage 2 permissions, they are first combined with any stage 1 permissions before applying the permission check.

**“combined-st1-st2”**

Always fetches all stages and combine before applying the permissions check.

Type: `string`

Default value: `""`

### **ordering\_of\_PAN\_and\_xn\_by\_ns\_pas**

Execution from ns-pas can be forbidden:

- For secure, this is controlled by `SMMU_S_CRO.SIF`.
- For realm, this is mandatory. In the model, we call this RIF and is always cached in the TLB.

When PAN is interpreted as EPAN then whether SIF/RIF is applied before or after PAN can get different results for the direct permission model.

In the model, the following options are available:

**0**

PAN applied first, SIF not cached in the TLB, RIF cached in the TLB.

**1**

PAN applied first, SIF/RIF cached in the TLB.

**2**

SIF/RIF cached in the TLB, then PAN applied.



If you cache SIF in the TLB then all SIF faults are no longer traced separately as SIF faults but as permission faults, which architecturally they are reported as. As RIF is always cached in the TLB, they are not distinguished in the trace separately from permission faults.

---

Type: `unsigned`

Default value: `0`

### **out\_of\_range\_CMD\_ATC\_INV\_Size**

If `CMD_ATC_INV.Size > 52`, the model is allowed to:

**0**

Raise `CERROR_ILL`

**1**

Treat as **NOP**

The architecture also allows for an **UNKNOWN** invalidate size to be used, but the model does not support this.

Type: `unsigned`

Default value: `0`

**out\_of\_range\_l0gptsz\_s**

If the port l0gptsz\_s is driven to an invalid value and that value is used then the following behaviors are possible:

**-2**

Report the incoming value in SMMU\_ROOT\_GPT\_BASE\_CFG.LOGPTSZ and all transactions report a GPT Config Error if GPC checking is enabled.

**-1**

Produce an error and make the model unusable (default).

**invalid LOGPTSZ encoding**

Report this value in SMMU\_ROOT\_GPT\_BASE\_CFG.LOGPTSZ and all transactions report a GPT Config Error if GPC checking is enabled.

All other values are reserved and act as -1.

Type: `int32_t`

Default value: -1

**output\_attribute\_transform**

Transform downstream attributes of StreamID transactions.



The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

- "" or "none": no transform
- How to alter output attributes. Example:

```
"ExtendedID[15:0]=DeviceID[15:0], UserFlags[31]=nSSV, UserFlags[19:0]=SubstreamID,
ManagerID64[10]=ManagerID64[11], ManagerID64[11]=ManagerID64[10]"
```

RHS/LHS Symbols:

**ExtendedID / ManagerID64 / UserFlags**

Incoming/Outgoing PVBUS transaction attributes

RHS Symbols:

**DeviceID**

`StreamID + translated_device_id_base`

**StreamID / SubstreamID / SEC\_SID / SEC\_SID\_bit\_1 / SIDV / SSV**

Architectural information. See parameter `howto_identify` for more information.

**nSEC\_SID / nSEC\_SID\_bit\_1 / nSSV / nSIDV**

Negative logic of above parameters. Different attributes are independent and can use negative or positive logic.

**St1PBHA / St2PBHA**

Page Based Hardware Attributes from leaf descriptors (zero if not used).

**STE\_IMPDEF1**

STE[127:116]

**HWATTR\_KIND\_0**

PBHA information in the format of MMU-700 and later

**ALIGNED\_IPA\_55\_12**

If LOG2\_IPA\_REGION\_SIZE != 0, the translation has an IPA region whose aligned start address is ALIGNED\_IPA\_55\_12 << 12. Thus, the IPA of the transaction is:

(ALIGNED\_IPA\_55\_12 << 12) | (PA & ((1ull << LOG2\_IPA\_REGION\_SIZE) - 1))

**LOG2\_IPA\_REGION\_SIZE**

If LOG2\_IPA\_REGION\_SIZE == 0, no stage 2 has been applied and the ALIGNED\_IPA\_55\_12 field is meaningless. Otherwise, a stage 2 has been applied, which means that this field is the log2 of the size of the IPA region size and the start of the region is ALIGNED\_IPA\_55\_12 << 12.

**Numeric Literals**

Any number. Ex: 0x1234

Any bits with no transform are unchanged.



Note

Pre-RME, the SEC\_SID *input* to the SMMUv3 was a single bit. RME added a second bit. In order for the model to be backwards-compatible, the SEC\_SID *symbol* in this parameter remains as a single bit and a new symbol SEC\_SID\_bit\_1 has to be used to refer to the second bit of the SEC\_SID *input*.



Note

- mpam\_attribute\_transform and mec\_attribute\_transform are applied in order after this.
- See also output\_attribute\_transform\_for\_NoStreamID for NoStreamID transactions.

Type: string

Default value: "ExtendedID[31:0]=DeviceID"

## output\_attribute\_transform\_for\_NoStreamID

**Note**

The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

Transform downstream attributes of NoStreamID transactions.

- "" or "none": no transform
- How to alter output attributes. Example:

```
"ExtendedID[15:0]=0, UserFlags[31]=1, UserFlags[19:0]=0,
ManagerID64[10]=ManagerID64[11],
ManagerID64[11]=ManagerID64[10] ManagerID64[9:6]=HWATTR_KIND_0"
```

RHS/LHS Symbols: \* ExtendedID/ManagerID64/UserFlags: incoming/outgoing attributes.

RHS Symbols: \* SIDV = 0, nSIDV = 1 (fixed values to indicate NoStreamID) \* PAS \* HWATTR\_KIND\_0

Any bits with no transform are unchanged.

**Note**

- mpam\_attribute\_transform and mec\_attribute\_transform are applied in order after this.
- see also output\_attribute\_transform for StreamID transactions.

Type: string

Default value: "ExtendedID[31:0]=0, ExtendedID[32]=1"

## output\_id\_routed\_transform

**Note**

The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

The SMMU generates the following ID-routed transaction on the pvbus\_id\_routed\_m bus:

- ATC Invalidate
- PRI Response

This parameter controls how the SMMU should express:

- the StreamID

- the Trusted (T) bit

The value is a comma-separated list of assignments:

```
Address[27:12]=StreamID[15:0], ExtendedID[60]=T, ExtendedID[15:0]=StreamID[31:16]
```

Address bits[11:0] cannot be used.

The LHS can be one of:

- PAS
- ManagerID64 / ExtendedID / UserFlags
- Address

The RHS can be one of:

- a numeric constant
- SSD
- T or negative version nT
- StreamID

For realm (or 'Trusted') transactions, then  $ssd=0b11$ ,  $T=1$ ,  $nT=0$ . For non-secure (or 'Non-Trusted') transactions, then  $ssd=0b01$ ,  $T=0$ ,  $nT=1$ .

Type: string

Default value: "Address[43:12]=StreamID, PAS=SSD"

### **percent\_commit**

Percentage of times that a read of a register with Update commits the update. 0 means commit immediately.

Type: uint32\_t

Default value: 20

### **percent\_commit\_Update\_clear**

Percentage of times that a read of a register with a pending Update clear lowers the Update flag.

Type: uint32\_t

Default value: 20

### **pmu**

What to instantiate as a PMU.



All events and counters are intended for demonstration purposes only and should not be treated as in any way reflecting accurate values for a real implementation. The model's internal representation of actions differs significantly from real hardware and the particular value obtained from the counters should not be used for benchmarking.

Values of this parameter are:

""

No PMU

#### "distributed-0"

- A PMCG per TBU (number\_of\_ports, up to 63 ports).
- A single PMCG for a TCU.
- Connect a debugger to see the configuration.

#### "distributed-1"

The same as distributed-0, except for supporting MSIs and MPAM on the MSIs if MPAM is supported by rest of the SMMU.

Type: `string`

Default value: ""

### **ports\_that\_ignore\_PnU\_InD\_on\_transactions\_with\_no\_SubstreamID**

Some bus systems, notably PCIe, do not support marking a transaction as Privileged/User or Instruction/Data unless the transaction has a SubstreamID.

This accepts a comma-separated list of numbers and ranges, for example:

```
0, 10-12, 15
```

If the number P is named in this list then the upstream `pvbus_s[P]` will have all transactions with no Substream considered to be User and Data.

Type: `string`

Default value: ""

### **prefetch\_only\_requests**

The simulator supports 'prefetch-only' DMI requests, which can occur at any time and for any reason and are intended to be invisible to the end execution of the model and to the user.

0

deny all prefetch-only requests



**1**

- use debug requests for any page table walks
  - form and use debug TLB/cache entries
  - any faults will not record, but deny the prefetch request

**2**

- treat prefetch-only requests like normal transactions
  - use normal page table walk transactions
  - use and form normal TLB/cache entries
  - faults will alter the programmer-visible state of the SMMU

0 is the safest.

1 treats the access like a debug request and requires that debug page table walks are treated correctly downstream. Any descriptors that need HTTU to allow the transaction to proceed will fail the request.

2 is dangerous, it uses real transactions and reports faults that are unphysical. Real transactions can be `wait()`ed and this disobeys the SystemC spec for `get_direct_mem_ptr()`.

Type: `unsigned`

Default value: 0

**ras**

What to instantiate for RAS handling.

Values of this parameter are:

- "" – no RAS records
- "MMU\_600"
- "MMU\_700"
- "MMU\_S3"

If "MMU\_600", only corrected errors are reported.

See also `imp_def_ras_allow_non_secure_accesses_if_supports_secure`.

Type: `string`

Default value: ""

**register\_accesses\_to\_root\_or\_realm\_pas\_when\_no\_rme**

When RME is not implemented or is disabled by `legacy_tz_en`:

**0**

Root and realm register PAS accesses are not **RAZ/WI**.

**1**

Root and realm register PAS accesses are treated as **RAZ/WI**.

Type: `uint32_t`

Default value: 0

**reset\_value\_of\_SMMU\_GBPA**

Reset value of SMMU\_GBPA.

Type: `uint32_t`

Default value: 0

**reset\_value\_of\_SMMU\_S\_GBPA**

Reset value of SMMU\_S\_GBPA.

Type: `uint32_t`

Default value: 0

**rme\_ats\_request\_pa\_strategy**

When RME\_IMPL == 0, the PA of an ATS Request's response is permitted but not required to undergo a GPT check:

**0**

Do not check the PA

**1**

Do the check against the PA

**2**

Check the PA 50% of the time

Translated transactions are required to always undergo a GPT check whatever happens.

This parameter is ignored if RME\_IMPL==1 and the PA is required to be checked.

Type: `uint32_t`

Default value: 0

**rme\_da\_force\_better\_configuration**

RME-DA requires that the SMMU be integrated into a system for which SMMU\_IDR0.COHAAC == 1 and SMMU\_IDR0.IDR0.HTTU == both\_af\_and\_dirty (2).

The model has the following pins which can control these ID fields:

- `conf_system_supports_cohacc`
- `conf_system_supports_httu`

In addition, RME-DA requires that the fundamental SMMU has certain properties that are configured by its ID codes.

This parameter allows you to selectively ignore the pins and bad ID to produce a good configuration by forcing the required values.

Whether the SMMU has RME-DA or not is identified by `SMMU_ROOT_IDR0.REALM_IMPL`.

This is a comma-separated list of fields to force when RME-DA is configured by `SMMU_ROOT_IDR0.REALM_IMPL`:

- In `SMMU_IDR0`:
  - "Hyp"
  - "S1P"
  - "S2P"
  - "TTF"
  - "NS1ATS"
  - "COHACC"
  - "HTTU"
  - "RME\_IMPL"
- "SSIDSIZE" in `SMMU_IDR1`
- "BBML" in `SMMU_IDR3`

You can also use "all" to set all.

Type: `string`

Default value: ""

### **`rme_10gpt_entry_covers_log2size_in_bytes`**

Each LOGPT entry covers:

```
2**rme_10gpt_entry_covers_log2size_in_bytes
```

bytes of address space.

The valid values for this parameter are: 30, 34, 36, 39

This parameter is reported in an encoded format as the read-only field:

```
SMMU_ROOT_GPT_BASE_CFG.LOGPTSZ
```

This parameter can be overridden by the port `10gptsz_s` when sampled on negedge of reset.

Type: `uint32_t`

Default value: 30

### **`rme_speculation_control`**

This is a comma-separated list of flags that control when and how the model performs speculation for RME.

Type: `string`

Default value: ""

### **`root_register_page_offset`**

This is the offset from `SMMU_BASE` of the Root register file page which is 64 KiB in size. It must not overlap any other part of the register map.

Type: `uint64_t`

Default value: 0

### **`secure_state_controls_access_to_SMMU_S_INIT`**

With RME, access control of the `SMMU_S_INIT` belongs to Root. This parameter allows Root to delegate access control to the secure state, enabling secure software to reset the TLB, clearing out any TLB entries.

If RME is implemented and this parameter is 0, `allow_non_secure_access_to_SMMU_S_INIT` has no effect.

If the SMMU does not implement RME, this parameter is ignored.

Type: `bool`

Default value: `true`

### **`seed`**

Used to seed the pseudo-random number generator that the SMMU model uses.

Type: `uint32_t`

Default value: 0x12345678

### **`separate_tw_msi_qs_port`**

True if there is a separate port which is used to walk configuration tables, translation tables, issue MSIs and access the queues. If this is false then `pvbus_m[0]` is used.

Type: `bool`

Default value: `true`

### **`size_of_cd_cache`**

The number of entries in the cache holding CD structures. If this is zero it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

Type: `uint32_t`

Default value: 0

### **`size_of_dpttlb`**

The number of entries in the DPT TLB. If this is zero then it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

Type: `uint32_t`

Default value: 0

### **`size_of_gpptlb`**

The number of entries in the GPT TLB. If this is zero then it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

Type: `uint32_t`

Default value: 0

### **`size_of_llcd_cache`**

The number of entries in the cache holding L1CD descriptors. If this is zero it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

Type: `uint32_t`

Default value: 0

### **`size_of_llste_cache`**

The number of entries in the cache holding L1STE descriptors. If this is zero it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

Type: `uint32_t`

Default value: 0

### **`size_of_register_file`**

This is the power of two size that the register file occupies in the memory map. It is used to generate a mask for the addresses received on `pvbus_control_s` to decode the desired register offset.

The default for this parameter is 1 MiB.

Type: `uint64_t`

Default value: `0x100000`

### **`size_of_ste_cache`**

The number of entries in the cache holding STE structures. If this is zero it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

Type: `uint32_t`

Default value: `0`

### **`size_of_tlb`**

The number of entries in the TLB. If this is zero it is treated as a large number ('infinite') but it is bounded so that the host memory usage of the cache is also bounded.

Type: `uint32_t`

Default value: `0`

### **`smmu_msi_device_id`**

When appropriately enabled, assume that MSIs that are generated by the SMMU are presented to the GIC with this DeviceID.

See parameters `msi_attribute_transform` and `enable_device_id_checks`.

Type: `uint32_t`

Default value: `0`

### **`smmuv33_begin_offset_of_qcp0`**

This is the offset from `SMMU_BASE` of the first QCP page. The architecture requires that if more than one world of QCPs are present then they are in the order non-secure and then secure QCPs and form one continuous address space in the register file.

Type: `uint32_t`

Default value: `N/A`

### **`support_for_httu_when_starts_disallowed`**

`SMMU_IDR0.HTTU` describes to the programmer whether the SMMU and system support HTTU. Typically, an SMMU that is capable of HTTU has a configuration pin that says whether the system supports HTTU or not.

The SMMU model determines `SMMU_IDR0.HTTU` as follows:

- If the parameter `SMMU_IDR0` indicates any kind of support for HTTU, then the configuration pin turns support on and off between that value and no support for HTTU.

- If the parameter `SMMU_IDR0` indicates no HTTU support, allow the pin to turn on support to that specified by this parameter.

Values for this parameter are the same as for the `SMMU_IDR0.HTTU` field:

**0**

No support for HTTU

**1**

AF flag only

**2**

AF flag and DBM update.

Type: unsigned

Default value: 0

### **`tlb_when_do_f_tlb_conflict_on_overlap`**

If a TLB entry is created by a walk and it overlaps an existing entry, there are some architectural situations where the result is known. For all others, then an implementation is allowed to use an **UNPREDICTABLE** combination of the two entries, or it can generate `F_TLB_CONFLICT`:

**0**

never generate

**1**

sometimes generate

**2**

always generate

Conflicts between global and non-global entries are not detected by the model.

Type: unsigned

Default value: 0

### **`translated_device_id_base`**

When appropriately enabled, assume that client device accesses are translated to a DeviceID as seen by the GIC of:

```
StreamID + translated_device_id_base
```

See parameter `output_attribute_transform` and `enable_device_id_checks`.

Type: `uint32_t`

Default value: 0

**treat\_debug\_read\_accesses\_as\_speculative\_accesses**

The SMMU architecture has the concept of speculative accesses. If you set this flag to true, debug read accesses flowing from the upstream system through the SMMU are interpreted as speculative.

The difference is that a speculative read:

- Participates in HTTU
- If it encounters a (non-HTTU) fault, always returns abort

Debug writes are still considered as debug accesses. All speculative writes would be aborted and this is not a useful behavior for the SMMU to emulate.

Type: `bool`

Default value: `false`

**tw\_qs\_attribute\_transform**

Transform downstream attributes of table walk and queue transactions.



The parameter is parsed according to the rules in the section 'Parameters for parsing transaction attributes'.

- "" or "none" – no transform
- How to alter the output attributes. Example:

```
"ExtendedID[35:32]=HWATTR_KIND_0"
```

RHS/LHS Symbols:

**ExtendedID / ManagerID64 / UserFlags**

Incoming/Outgoing PVBUS transaction attributes

RHS Symbols:

**HWATTR\_KIND\_0**

PBHA information

**kind**

Transaction kind

- For a read:

**0/1**

L1STE/STE

**2/3**

L1CD/CD



**4/5**

S1/S2 TTD (including CAS)

**6**

CMDQ

**7**

VMS

**11/12**

LOGPT/L1GPT

**13/14**

LODPT/L1DPT

- For a write:

**0**

EVENTQ

**1**

PRIQ

ExtendedID/ManagerID64/UserFlags start with values {0, 0xFFFFffff, 0} respectively.

Any bits with no transform are unchanged.



See also `output_attribute_transform` and `msi_attribute_transform`.

---

Type: string

Default value: ""

### **unpred\_httu\_percent\_do\_discretionary\_AF**

If a descriptor could have a discretionary update of the AF flag on, what is the percentage of the time that the AF update should occur.

Type: unsigned

Default value: 50

### **unpred\_httu\_percent\_do\_discretionary\_DBM**

If a descriptor could have a discretionary DBM update to make the descriptor WriteableDirty, what is the percentage of the time that the DBM update should occur.

Type: unsigned

Default value: 50

**unpred\_translated\_access\_out\_of\_range\_of\_oas**

If a Translated Access is presented to the SMMU that is > OAS then it is CONSTRAINED UNPRED as to whether the transaction will either:

**0**

Be truncated to OAS and go downstream

**1**

Be aborted, no event written.

Type: unsigned

Default value: 1

**wait\_atos\_ticks**

This is the time to wait before doing an ATOS operation. If bit 32 is set (0x1\_0000\_0000) then the time waited for is a uniform randomly distributed time  $[0, \max(2, (t \ \& \ 0xFFFFFFFF)) - 1]$ .

Type: uint64\_t

Default value: 0

**wait\_cmdq\_ticks**

This is the time to wait before doing something on the command queue. If bit [32] is set 0x1\_0000\_0000 then the time waited for is a uniform randomly-distributed time  $[0, \max(2, (t \ \& \ 0xFFFFFFFF)) - 1]$ .

Type: uint64\_t

Default value: 0

**wait\_eventq\_ticks**

This is the time to wait before doing something on the event queue. If bit [32] is set 0x1\_0000\_0000 then the time waited for is a uniform randomly-distributed time  $[0, \max(2, (t \ \& \ 0xFFFFFFFF)) - 1]$ .

Type: uint64\_t

Default value: 0

**wait\_misc\_async\_actions\_ticks**

This is the time to wait before doing an async action that could be delayed is performed. If bit [32] is set 0x1\_0000\_0000 then the time waited for is a uniform randomly-distributed time  $[0, \max(2, (t \ \& \ 0xFFFFFFFF)) - 1]$ .

Type: uint64\_t

Default value: 0

**wait\_msi\_ticks**

This is the time to wait before sending an MSI. If bit [32] is set 0x1\_0000\_0000 then the time waited for is a uniform randomly-distributed time  $[0, \max(2, (t \ \& \ 0xFFFFFFFF)) - 1]$ .

Type: uint64\_t

Default value: 0

**wait\_pri\_req\_ticks**

This is the time to wait before processing a PRI Request. If bit [32] is set 0x1\_0000\_0000 then the time waited for is a uniform randomly-distributed time  $[0, \max(2, (t \ \& \ 0xFFFFFFFF)) - 1]$ .

Type: uint64\_t

Default value: 0

**wait\_pri\_resp\_ticks**

This is the time to wait before sending a PRI Response back to the PCIe subsystem. When a PRI Response is an auto-response then the ATC might immediately make a new ATS request, that immediately fails, immediately makes a PRI Request, or auto-responds, etc. To break this loop, then we introduce a minimum time on all PRI Responses to give other components in the system a chance to run. If bit [32] is set 0x1\_0000\_0000 then the time waited for is a uniform randomly-distributed time  $[0, \max(2, (t \ \& \ 0xFFFFFFFF)) - 1]$ .

Type: uint64\_t

Default value: 1

**when\_fetch\_vms**

Architecturally, there is flexibility in how a VMS is cached and thus:

- When it will be fetched.
- The prioritization of F\_VMS\_FETCH.

Of the many architecturally-allowed options, the model offers two:

**0**

    Fetched and cached immediately after the STE is fetched

**1**

    Fetched and cached immediately after the CD is fetched

In both cases, the VMS is cached in the STE and CMD\_CFGI\_VMS\_PIDM is a **NOP**.

Type: unsigned

Default value: 0

**width\_of\_agbpa\_impdef**  
Width of the SMMU\_s\_AGBPA.IMPDEF field.

Type: uint32\_t

Default value: 16

3.375 SMMUv3AEMIdentify2AMBAPVValue64

Defined in examples/SystemCExport/Bridges/SMMUv3AEMIdentify2AMBAPVValue64.lisa.

**About SMMUv3AEMIdentify2AMBAPVValue64**  
SMMUv3AEMIdentify to AMBA-PV Value64 protocol converter.

**Iris and MTI instances for SMMUv3AEMIdentify2AMBAPVValue64**  
This model has the following Iris instances:

Name	Instance type
SMMUv3AEMIdentify2AMBAPVValue64	SMMUv3AEMIdentify2AMBAPVValue64

Ports for SMMUv3AEMIdentify2AMBAPVValue64

Port	Direction	Protocol	Description
identify_reply	slave	AMBAPVValue64	From SystemC.
identify_request	master	AMBAPVValue64	To SystemC.
identify	slave	SMMUv3AEMIdentifyProtocol	SMMUv3AEMIdentifyProtocol input.

**Parameters for SMMUv3AEMIdentify2AMBAPVValue64**  
This component does not have any parameters.

3.376 SMMUv3TestEngine

Defined in LISA/SMMUv3TestEngine.lisa.

**About SMMUv3TestEngine**  
Test Engine used for testing SMMUv3.

**Iris and MTI instances for SMMUv3TestEngine**  
This model has the following Iris instances:

Name	Instance type
SMMUv3TestEngine	SMMUv3TestEngine
SMMUv3TestEngine.register_file[0]	PVBusSlave

This model has the following MTI trace components:

Name	Component type
SMMUv3TestEngine	SMMUv3TestEngine
SMMUv3TestEngine.register_file[0]	PVBusSlave

### Ports for SMMUv3TestEngine

Port	Direction	Protocol	Description
client_s	slave	PCIDevice2ClientProtocol	-
clk_in	slave	ClockSignal	-
identify	slave	SMMUv3AEMIdentifyProtocol	-
pvbus_control_s	slave	PVBus	-
pvbus_m	master	PVBus	-
reset_in	slave	Signal	-

### Parameters for SMMUv3TestEngine

#### **bandwidth\_per\_transaction\_in\_bytes\_per\_tick**

The bandwidth of the device for each in-flight transaction, in bytes/tick of clk\_in. This is only a rough guess. If you are uninterested in trying to run cores and the engine simultaneously then set this to a large number.

Type: uint32\_t

Default value: 100

#### **max\_number\_of\_inflight\_transactions**

The maximum number of in-flight transactions allowed.

Type: uint32\_t

Default value: 10

#### **output\_attribute\_transform**

How to pack the stream identification information into the transaction attributes.is:-<empty> or "default""pcie" the de-facto standard for the PCIe subsystem in FastModels<empty> or "default" is equivalent to:-ExtendedID[63]=nSEC\_SID, ExtendedID[55:24]=StreamID, ExtendedID[20]=nSSV, ExtendedID[19:0]=SubstreamID"pcie" option is equivalent to:-ExtendedID[63]=SEC\_SID, ExtendedID[62]=SSV, ExtendedID[51:32]=SubstreamID, ExtendedID[31:0]=StreamID.

Type: string

Default value: ""

#### **seed**

The seed used for various random number generators.

Type: `uint32_t`

Default value: `0x12345678`

## 3.377 SMSC\_91C111

Defined in `LISA/SMSC_91C111.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### About SMSC\_91C111

This component provides the register interface of the SMSC part and can be configured to act as an unconnected Ethernet port, or an Ethernet port connected to the host by an Ethernet bridge.

It uses a banked register model of primarily 16-bit registers. There are also indirectly accessible registers for the PHY unit.

If a MAC address is not specified in the `mac_address` parameter, the simulator takes the default MAC address, which is randomly generated. This provides some degree of MAC address uniqueness when running models on multiple hosts on a local network.



Note

DHCP servers allocate the IP addresses, but because they sometimes do this based on the MAC address provided to them, using random MAC addresses might interact unfortunately with some DHCP servers.

See also:

- [Configuring the networking environment for Linux](#)

### Iris and MTI instances for SMSC\_91C111

This model has the following Iris instances:

Name	Instance type
<code>SMSC_91C111</code>	<code>SMSC_91C111</code>
<code>SMSC_91C111.SMSC_slave</code>	<code>PVBusSlave</code>

This model has the following MTI trace components:

Name	Component type
<code>SMSC_91C111.SMSC_slave</code>	<code>PVBusSlave</code>

## Ports for SMSC\_91C111

Port	Direction	Protocol	Description
clock	slave	ClockSignal	Clock input, typically 25MHz, which sets the master transmit/receive rate.
eth	master	VirtualEthernet	Ethernet port.
intr	master	Signal	Interrupt signal.
pvbuse	slave	PVBus	Slave port for register access.
state	master	ValueState_64	State port to retrieve state of host bridge

## Parameters for SMSC\_91C111

### cache\_size

Size of cache memory in SMSC MMU.

Type: `int`

Default value: `0x10000`

### enabled

Host interface connection enabled.

Type: `bool`

Default value: `false`

### mac\_address

Host/model MAC address.

Type: `string`

Default value: `"00:02:f7:ef:00:00"`

### not\_lan911x

Gracefully fail SMSC LAN911x driver probe.

Type: `bool`

Default value: `false`

### promiscuous

Put host into promiscuous mode.

Type: `bool`

Default value: `true`

## 3.378 SP804\_Timer

Defined in `LISA/SP804_Timer.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r1p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### About SP804\_Timer

ARM Dual-Timer Module(SP804).

### Iris and MTI instances for SP804\_Timer

This model has the following Iris instances:

Name	Instance type
SP804_Timer	SP804_Timer
SP804_Timer.bussubordinate	PVBusSlave
SP804_Timer.clk_divY (where Y = 0-1)	ClockDivider
SP804_Timer.counterY (where Y = 0-1)	CounterModule
SP804_Timer.counterY.bussubordinate (where Y = 0-1)	PVBusSlave
SP804_Timer.counterY_bus_manager (where Y = 0-1)	PVBusMaster

This model has the following MTI trace components:

Name	Component type
SP804_Timer.bussubordinate	PVBusSlave
SP804_Timer.clk_divY (where Y = 0-1)	ClockDivider
SP804_Timer.counterY.bussubordinate (where Y = 0-1)	PVBusSlave
SP804_Timer.counterY_bus_manager (where Y = 0-1)	PVBusMaster

### Ports for SP804\_Timer

Port	Direction	Protocol	Description
clock	slave	<a href="#">ClockSignal</a>	Clock input, typically 1MHz, driving master count rate.
irq_out0	master	<a href="#">Signal</a>	Interrupt signaling.
irq_out1	master	<a href="#">Signal</a>	Interrupt signaling.
pvbus	slave	<a href="#">PVBus</a>	Subordinate port for register access.
timer_en	slave	<a href="#">ClockRateControl</a>	Port for changing the rate of timer n.

### Parameters for SP804\_Timer

This component does not have any parameters.



## 3.379 SP805\_Watchdog

Defined in `LISA/SP805_Watchdog.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r1p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### About SP805\_Watchdog

ARM Watchdog Module(SP805).

### Iris and MTI instances for SP805\_Watchdog

This model has the following Iris instances:

Name	Instance type
SP805_Watchdog	SP805_Watchdog
SP805_Watchdog.busslave	PVBusSlave
SP805_Watchdog.clocktimer	ClockTimerThread
SP805_Watchdog.clocktimer.timer	ClockTimerThread64
SP805_Watchdog.clocktimer.timer.thread	SchedulerThread
SP805_Watchdog.clocktimer.timer.thread_event	SchedulerThreadEvent

This model has the following MTI trace components:

Name	Component type
SP805_Watchdog.busslave	PVBusSlave

### Ports for SP805\_Watchdog

Port	Direction	Protocol	Description
clk_in	slave	ClockSignal	-
irq_out	master	Signal	-
pvbus_s	slave	PVBus	-
reset_in	slave	Signal	-
reset_out	master	Signal	-

### Parameters for SP805\_Watchdog

#### **simhalt**

Halt on reset.

Type: `bool`

Default value: `false`

## 3.380 SP810\_SysCtrl

Defined in `LISA/SP810_SysCtrl.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### Changes in 11.31.15

The following parameters were added:

- `verbosity`

### About SP810\_SysCtrl

PrimeXsys System Controller(SP810) NB: Only EB relevant functionalities are fully implemented.

### Iris and MTI instances for SP810\_SysCtrl

This model has the following Iris instances:

Name	Instance type
<code>SP810_SysCtrl</code>	<code>SP810_SysCtrl</code>
<code>SP810_SysCtrl.busslave</code>	<code>PVBusSlave</code>
<code>SP810_SysCtrl.clkdiv_clkY</code> (where $Y = 0-3$ )	<code>ClockDivider</code>

This model has the following MTI trace components:

Name	Component type
<code>SP810_SysCtrl.busslave</code>	<code>PVBusSlave</code>
<code>SP810_SysCtrl.clkdiv_clkY</code> (where $Y = 0-3$ )	<code>ClockDivider</code>

### Ports for SP810\_SysCtrl

Port	Direction	Protocol	Description
<code>clk_in</code>	slave	<code>ClockSignal</code>	-
<code>hclkdivsel</code>	master	<code>ValueState</code>	-
<code>npwr</code>	slave	<code>Signal</code>	-
<code>pll_en</code>	master	<code>Signal</code>	-
<code>pvbuss</code>	slave	<code>PVBus</code>	-

Port	Direction	Protocol	Description
ref_clk_in	slave	ClockSignal	-
remap_clear	master	StateSignal	-
remap_stat	slave	StateSignal	-
sleep_mode	master	Signal	-
sys_id	slave	ValueState	-
sys_mode	slave	ValueState	-
sys_stat	slave	ValueState	-
timer_clk_en	master	ClockRateControl	-
wd_clk_en	master	Signal	-
wd_en	slave	Signal	-

## Parameters for SP810\_SysCtrl

### **clkdiv\_clk0.div**

Clock Rate Divider.

Type: uint64\_t

Default value: 1

### **clkdiv\_clk0.mul**

Clock Rate Multiplier.

Type: uint64\_t

Default value: 1

### **clkdiv\_clk1.div**

Clock Rate Divider.

Type: uint64\_t

Default value: 1

### **clkdiv\_clk1.mul**

Clock Rate Multiplier.

Type: uint64\_t

Default value: 1

### **clkdiv\_clk2.div**

Clock Rate Divider.

Type: uint64\_t

Default value: 1

**clkdiv\_clk2.mul**

Clock Rate Multiplier.

Type: uint64\_t

Default value: 1

**clkdiv\_clk3.div**

Clock Rate Divider.

Type: uint64\_t

Default value: 1

**clkdiv\_clk3.mul**

Clock Rate Multiplier.

Type: uint64\_t

Default value: 1

**sysid**

System Identification Register.

Type: uint32\_t

Default value: 0x00000000

**use\_s8**

Use Switch 8 (S1-S4).

Type: bool

Default value: false

**verbosity**

Adds more logging: -1: print ERROR messages only, 0: print up to WARNING, 1: up to INFO, 2: up to DEBUG.

Type: int32\_t

Default value: 0

## 3.381 SSU

Defined in `LISA/ssu.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
rOp0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### About SSU

Safety Status Unit.

### Iris and MTI instances for SSU

This model has the following Iris instances:

Name	Instance type
SSU	SSU
SSU.pvbus_slave	PVBusSlave

This model has the following MTI trace components:

Name	Component type
SSU	SSU
SSU.pvbus_slave	PVBusSlave

### Ports for SSU

Port	Direction	Protocol	Description
c_error_in	slave	Signal	Critical interrupts to the SSU
clk_in	slave	ClockSignal	Clock input
cold_reset_in	slave	Signal	Cold reset signal
nc_error_in	slave	Signal	Non critical interrupts to the SSU
pvbus_s	slave	PVBus	To access FMU model registers
ssu_out	master	ValueState	SSU output port
warm_reset_in	slave	Signal	Warm reset signal

### Parameters for SSU

#### diagnostics

Diagnostics. 0: FatalError, 1:Error, 2:Warning, 3:Info, 4:Debug.

Type: `uint8_t`

Default value: 2

**sm\_implemented**

Safety Mechanism Implemented.

Type: `bool`

Default value: `false`

## 3.382 STLBusGasket

Defined in `LISA/STLBusGasket.lisa`.

### About STLBusGasket

STLBusGasket allows a debugger or emulated T32 code to force the results of system-register reads by writing an address to the ADDR register then 32-bit values to the VALUE register, which are placed in a fifo associated with that address. A PVBUS transaction into `pvbus_in` goes unchanged to `pvbus_out`, unless its address matches that associated with a non-empty fifo, in which case: writes are ignored, non-word reads abort, and word reads take values from that fifo.

### Iris and MTI instances for STLBusGasket

This model has the following Iris instances:

Name	Instance type
<code>STLBusGasket</code>	<a href="#">STLBusGasket</a>
<code>STLBusGasket.busmapper</code>	<a href="#">PVBusMapper</a>

This model has the following MTI trace components:

Name	Component type
<code>STLBusGasket.busmapper</code>	<a href="#">PVBusMapper</a>

### Ports for STLBusGasket

Port	Direction	Protocol	Description
<code>pvbus_in</code>	slave	<a href="#">PVBUS</a>	-
<code>pvbus_out</code>	master	<a href="#">PVBUS</a>	-

### Parameters for STLBusGasket

**function**

Function: 0-none, 1-STL value-forcing.

Type: `uint64_t`

Default value: 0

**reg\_base**

Base Address of STL control regs (ADDR,VAL at offsets 0,4).

Type: uint32\_t

Default value: 0xE001E820

**verbose**

Verbosity : 0-none, 1-some.

Type: uint64\_t

Default value: 0

## 3.383 ScalableClockControl

Defined in LISA/ScalableClockControl.lisa.

### About ScalableClockControl

Clock control allows input selection, rate control and gating.

### Iris and MTI instances for ScalableClockControl

This model has the following Iris instances:

Name	Instance type
ScalableClockControl	ScalableClockControl
ScalableClockControl.clkDivX (where X = 1-10)	ClockDivider
ScalableClockControl.clkGate	ClockGate
ScalableClockControl.clkGate.divider	ClockDivider
ScalableClockControl.clkSelect	ClockSelector
ScalableClockControl.clkSelect.clkdivX (where X = 0-10)	ClockDivider
ScalableClockControl.clkSelect.clkdivider	ClockDivider

This model has the following MTI trace components:

Name	Component type
ScalableClockControl.clkDivX (where X = 1-10)	ClockDivider
ScalableClockControl.clkGate.divider	ClockDivider
ScalableClockControl.clkSelect.clkdivX (where X = 0-10)	ClockDivider
ScalableClockControl.clkSelect.clkdivider	ClockDivider

## Ports for ScalableClockControl

Port	Direction	Protocol	Description
clk_out	master	ClockSignal	-
clkEnable	slave	Signal	-
clkSel	slave	Value	-
clock_in	slave	ClockSignal	-
clock_rate	slave	ClockRateControl	-
halt	master	Signal	-
refClk_in	slave	ClockSignal	-

## Parameters for ScalableClockControl

### **clkDiv1.div**

Clock Rate Divider.

Type: uint64\_t

Default value: 1

### **clkDiv1.mul**

Clock Rate Multiplier.

Type: uint64\_t

Default value: 1

### **clkDiv10.div**

Clock Rate Divider.

Type: uint64\_t

Default value: 1

### **clkDiv10.mul**

Clock Rate Multiplier.

Type: uint64\_t

Default value: 1

### **clkDiv2.div**

Clock Rate Divider.

Type: uint64\_t

Default value: 1



**clkDiv2.mul**

Clock Rate Multiplier.

Type: uint64\_t

Default value: 1

**clkDiv3.div**

Clock Rate Divider.

Type: uint64\_t

Default value: 1

**clkDiv3.mul**

Clock Rate Multiplier.

Type: uint64\_t

Default value: 1

**clkDiv4.div**

Clock Rate Divider.

Type: uint64\_t

Default value: 1

**clkDiv4.mul**

Clock Rate Multiplier.

Type: uint64\_t

Default value: 1

**clkDiv5.div**

Clock Rate Divider.

Type: uint64\_t

Default value: 1

**clkDiv5.mul**

Clock Rate Multiplier.

Type: uint64\_t

Default value: 1

**clkDiv6.div**

Clock Rate Divider.

Type: uint64\_t

Default value: 1

**clkDiv6.mul**

Clock Rate Multiplier.

Type: uint64\_t

Default value: 1

**clkDiv7.div**

Clock Rate Divider.

Type: uint64\_t

Default value: 1

**clkDiv7.mul**

Clock Rate Multiplier.

Type: uint64\_t

Default value: 1

**clkDiv8.div**

Clock Rate Divider.

Type: uint64\_t

Default value: 1

**clkDiv8.mul**

Clock Rate Multiplier.

Type: uint64\_t

Default value: 1

**clkDiv9.div**

Clock Rate Divider.

Type: uint64\_t

Default value: 1

**clkDiv9.mul**

Clock Rate Multiplier.

Type: uint64\_t

Default value: 1

**clkGate.diagnostics**

Diagnostics.

Type: uint32\_t

Default value: 0

**clkGate.divider.div**

Clock Rate Divider.

Type: uint64\_t

Default value: 1

**clkGate.divider.mul**

Clock Rate Multiplier.

Type: uint64\_t

Default value: 1

**clkSelect.clkdiv0.div**

Clock Rate Divider.

Type: uint64\_t

Default value: 1

**clkSelect.clkdiv0.mul**

Clock Rate Multiplier.

Type: uint64\_t

Default value: 1

**clkSelect.clkdiv1.div**

Clock Rate Divider.

Type: uint64\_t

Default value: 1

**clkSelect.clkdiv1.mul**

Clock Rate Multiplier.

Type: uint64\_t

Default value: 1

**clkSelect.clkdiv10.div**

Clock Rate Divider.

Type: uint64\_t

Default value: 1

**clkSelect.clkdiv10.mul**

Clock Rate Multiplier.

Type: uint64\_t

Default value: 1

**clkSelect.clkdiv2.div**

Clock Rate Divider.

Type: uint64\_t

Default value: 1

**clkSelect.clkdiv2.mul**

Clock Rate Multiplier.

Type: uint64\_t

Default value: 1

**clkSelect.clkdiv3.div**

Clock Rate Divider.

Type: uint64\_t

Default value: 1

**clkSelect.clkdiv3.mul**

Clock Rate Multiplier.

Type: uint64\_t

Default value: 1

**clkSelect.clkdiv4.div**

Clock Rate Divider.

Type: uint64\_t

Default value: 1

**clkSelect.clkdiv4.mul**

Clock Rate Multiplier.

Type: uint64\_t

Default value: 1

**clkSelect.clkdiv5.div**

Clock Rate Divider.

Type: uint64\_t

Default value: 1

**clkSelect.clkdiv5.mul**

Clock Rate Multiplier.

Type: uint64\_t

Default value: 1

**clkSelect.clkdiv6.div**

Clock Rate Divider.

Type: uint64\_t

Default value: 1

**clkSelect.clkdiv6.mul**

Clock Rate Multiplier.

Type: uint64\_t

Default value: 1

**clkSelect.clkdiv7.div**

Clock Rate Divider.

Type: uint64\_t

Default value: 1

**clkSelect.clkdiv7.mul**

Clock Rate Multiplier.

Type: uint64\_t

Default value: 1

**clkSelect.clkdiv8.div**

Clock Rate Divider.

Type: uint64\_t

Default value: 1

**clkSelect.clkdiv8.mul**

Clock Rate Multiplier.

Type: uint64\_t

Default value: 1

**clkSelect.clkdiv9.div**

Clock Rate Divider.

Type: uint64\_t

Default value: 1

**clkSelect.clkdiv9.mul**

Clock Rate Multiplier.

Type: uint64\_t

Default value: 1

**clkSelect.clkdivider.div**

Clock Rate Divider.

Type: uint64\_t

Default value: 1

**clkSelect.clkdivider.mul**

Clock Rate Multiplier.

Type: uint64\_t

Default value: 1

**clkSelect.diagnostics**

Diagnostics.

Type: uint32\_t

Default value: 0

## 3.384 SchedulerInterface

Defined in LISA/SchedulerInterface.lisa.

### About SchedulerInterface

A SchedulerInterface instance allows access to the Fast Models scheduler.

### Iris and MTI instances for SchedulerInterface

This model has the following Iris instances:

Name	Instance type
SchedulerInterface	SchedulerInterface

### Ports for SchedulerInterface

Port	Direction	Protocol	Description
clk_in	slave	ClockSignal	Clock frequency for waitTicks() function.
control	slave	SchedulerInterfaceControl	Scheduler interface. Allows to: - wait for time

### Parameters for SchedulerInterface

This component does not have any parameters.

## 3.385 SchedulerThread

Defined in LISA/SchedulerThread.lisa.

### About SchedulerThread

A SchedulerThread instance represents a co-routine thread in the simulation.

### Iris and MTI instances for SchedulerThread

This model has the following Iris instances:

Name	Instance type
SchedulerThread	SchedulerThread

### Ports for SchedulerThread

Port	Direction	Protocol	Description
clk_in	slave	<a href="#">ClockSignal</a>	Clock frequency for waitTicks() function.
control	slave	<a href="#">SchedulerThreadControl</a>	SchedulerThread control. Managers use this to: - control the thread (wait etc) - implement the actual thread function threadProc;

### Parameters for SchedulerThread

This component does not have any parameters.

## 3.386 SchedulerThreadEvent

Defined in `LISA/SchedulerThreadEvent.lisa`.

### About SchedulerThreadEvent

A `SchedulerThreadEvent` instance is an auto-reset boolean condition variable other threads can wait on.

### Iris and MTI instances for SchedulerThreadEvent

This model has the following Iris instances:

Name	Instance type
<code>SchedulerThreadEvent</code>	<a href="#">SchedulerThreadEvent</a>

### Ports for SchedulerThreadEvent

Port	Direction	Protocol	Description
control	slave	<a href="#">SchedulerThreadEventControl</a>	-

### Parameters for SchedulerThreadEvent

This component does not have any parameters.

## 3.387 SecureAlarmManager

Defined in `LISA/SecureAlarmManager.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
0.93	Alpha support

For an explanation of the quality levels, see [Quality level definitions](#).



## Changes in 11.31.15

The following parameters were added:

- `disable_samrl_mask`
- `samem0_reset_val`
- `samem1_reset_val`
- `samim0_reset_val`
- `samim1_reset_val`
- `samrrls0_reset_val`
- `samrrls1_reset_val`

## About SecureAlarmManager

Security Alarm Manager.

## Iris and MTI instances for SecureAlarmManager

This model has the following Iris instances:

Name	Instance type
SecureAlarmManager	SecureAlarmManager
SecureAlarmManager.apb	PVBusSlave

This model has the following MTI trace components:

Name	Component type
SecureAlarmManager.apb	PVBusSlave

## Ports for SecureAlarmManager

Port	Direction	Protocol	Description
apb	slave	PVBus	APB Subordinate Interface - Access to registers
clk_in	slave	ClockSignal	Clock in signal
config_done_trig_ack_in	slave	Signal	Config done ack signal
config_done_trig_req_out	master	Signal	Config done req signal
double_bit_ecc_addr_in	slave	Value	Double bit ECC error address
event_in	slave	Signal	Event in signal
event_status_out	master	Signal	Event status out signal
nCOLDRESETAON_in	slave	Signal	Coldreset in signal
parwrite_addr_in	slave	Value	Partial Write Address
reset_in	slave	Signal	Reset in signal
response_action_out	master	Signal	Response action out signal
single_bit_ecc_addr_in	slave	Value	Single bit ECC error address

## Parameters for SecureAlarmManager

### **NUM\_SAMNEC**

Number of SAM event counters.

Type: `uint32_t`

Default value: 3

### **NUM\_SAMNRA**

Number of SAM response actions.

Type: `uint32_t`

Default value: 7

### **diagnostics**

Diagnostics.

Type: `uint32_t`

Default value: 0

### **disable\_samicv\_check**

Disable SAMICV check.

Type: `bool`

Default value: `false`

### **disable\_samrl\_mask**

Disable SAMRL read mask.

Type: `bool`

Default value: `false`

### **samem0\_reset\_val**

SAMEM0 reset value.

Type: `uint32_t`

Default value: `0x80D`

### **samem1\_reset\_val**

SAMEM1 reset value.

Type: uint32\_t

Default value: 0x0

**samim0\_reset\_val**

SAMIM0 reset value.

Type: uint32\_t

Default value: 0xFFFFFFFF

**samim1\_reset\_val**

SAMIM1 reset value.

Type: uint32\_t

Default value: 0x1F

**samrrls0\_reset\_val**

SAMRRLS0 reset value.

Type: uint32\_t

Default value: 0x8828

**samrrls1\_reset\_val**

SAMRRLS1 reset value.

Type: uint32\_t

Default value: 0x8000

3.388 SecureCache

Defined in LISA/SecureICache.lisa.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
1.13	Alpha support

For an explanation of the quality levels, see [Quality level definitions](#).

About SecureCache

SecureCache.

## Iris and MTI instances for SecureICache

This model has the following Iris instances:

Name	Instance type
SecureICache	SecureICache
SecureICache.DECRYPT_RAM_X (where X = 0-1)	PVBusSlave
SecureICache.HTR_RAM	PVBusSlave
SecureICache.Internal Master[0].pvbusmaster	PVBusMaster
SecureICache.SIC_BusMapper	PVBusMapper
SecureICache.SIC_BusMapperX (where X = 0-1)	PVBusMapper
SecureICache.ZERO_RAM	PVBusSlave
SecureICache.apb	PVBusSlave

This model has the following MTI trace components:

Name	Component type
SecureICache.DECRYPT_RAM_X (where X = 0-1)	PVBusSlave
SecureICache.HTR_RAM	PVBusSlave
SecureICache.Internal Master[0].pvbusmaster	PVBusMaster
SecureICache.SIC_BusMapper	PVBusMapper
SecureICache.SIC_BusMapperX (where X = 0-1)	PVBusMapper
SecureICache.ZERO_RAM	PVBusSlave
SecureICache.apb	PVBusSlave

## Ports for SecureICache

Port	Direction	Protocol	Description
apb	slave	PVBus	APB4 Subordinate Interface - Access to registers
htr_ram_pvbus_s	slave	PVBus	To Read and Write pre-calculated SHA-256 Digests
irq_out	master	Signal	To indicate interrupts
pvbus_m	master	PVBus	To read from External RAM
pvbus_s	slave	PVBus	Receives Transactions from CPU
reset_in	slave	Signal	reset

## Parameters for SecureICache

### **SIC\_AUTH\_ENABLE**

SIC Authentication enabled.

Type: `bool`

Default value: `false`

**SIC\_DECRYPT\_ENABLE**

SIC Decryption enabled.

Type: `bool`

Default value: `false`

**SIC\_DR\_CNT**

SIC config: Decryption Region Count (0x0=1, 0x1=2, 0x2=4).

Type: `uint32_t`

Default value: `0x1`

**SIC\_HTR\_RAM\_SIZE**

SIC config: Hash Tag RAM Size (0x1=1KB, 0x2=2KB, 0x4=4KB, 0x8=8KB, 0x10=16KB, 0x20=32KB).

Type: `uint32_t`

Default value: `0x20`

**SIC\_MAX\_CODE\_SIZE**

SIC config: Maximum Code Size in KB.

Type: `uint32_t`

Default value: `1024`

**SIC\_PAGE\_RAM\_SIZE**

SIC config: Page RAM Size (0x4=4KB, 0x8=8KB, 0x10=16KB).

Type: `uint32_t`

Default value: `0x10`

**SIC\_PAGE\_SIZE**

SIC config: Page Size (0x0=128B, 0x1=256B, 0x2=512B, 0x3=1KB, 0x4=2KB, 0x5=4KB).

Type: `uint32_t`

Default value: `0x3`

**SIC\_PMON\_EN**

SIC config: Performance Monitor enable.

Type: `bool`

Default value: `false`

**diagnostics**

Diagnostics.

Type: `uint32_t`

Default value: 2

3.389 SerialCrossover

Defined in `LISA/SerialCrossover.lisa`.

About SerialCrossover

This component implements two `SerialData` slave ports and can connect two `SerialData` master ports, such as from `PL011_Uart` components. Data received on one port is buffered in a FIFO until it is read from the other port. Signals received on one port are latched and available to be read by the other port.

Iris and MTI instances for SerialCrossover

This model has the following Iris instances:

Name	Instance type
SerialCrossover	<a href="#">SerialCrossOver</a>

Ports for SerialCrossover

Port	Direction	Protocol	Description
<code>port_a</code>	slave	<a href="#">SerialData</a>	Slave port for connecting to a <code>SerialData</code> master.
<code>port_b</code>	slave	<a href="#">SerialData</a>	Slave port for connecting to a <code>SerialData</code> master.

Parameters for SerialCrossover

**buffer\_capacity**

Buffer size for the UART FIFO (default: 32).

Type: `int`

Default value: 32

**diagnostics**

Diagnostics.

Type: `uint8_t`

Default value: 0

## 3.390 SignalDriver

Defined in `LISA/SignalDriver.lisa`.

### About SignalDriver

Drives signal port based on parameter, register or bus slave port.

### Iris and MTI instances for SignalDriver

This model has the following Iris instances:

Name	Instance type
SignalDriver	SignalDriver
SignalDriver.pvbuslave	PVBusSlave

This model has the following MTI trace components:

Name	Component type
SignalDriver	SignalDriver
SignalDriver.pvbuslave	PVBusSlave

### Ports for SignalDriver

Port	Direction	Protocol	Description
pvbus_s	slave	PVBus	-
signal_out	master	Signal	-

### Parameters for SignalDriver

#### **param\_input**

Drive signal\_out port with this parameter value.

Type: `bool`

Default value: `false`

## 3.391 SignalInverter

Defined in `LISA/SignalInverter.lisa`.

### Iris and MTI instances for SignalInverter

This model has the following Iris instances:

Name	Instance type
SignalInverter	SignalInverter

### Ports for SignalInverter

Port	Direction	Protocol	Description
sig_in	slave	Signal	-
sig_out_invert	master	Signal	-
sig_out	master	Signal	-

### Parameters for SignalInverter

This component does not have any parameters.

## 3.392 SignalLogger

Defined in LISA/SignalLogger.lisa.

### About SignalLogger

Traces signal activity.

### Iris and MTI instances for SignalLogger

This model has the following Iris instances:

Name	Instance type
SignalLogger	SignalLogger

This model has the following MTI trace components:

Name	Component type
SignalLogger	SignalLogger

### Ports for SignalLogger

Port	Direction	Protocol	Description
in	slave	Signal	Input signal port.
out	master	Signal	Output signal port.

### Parameters for SignalLogger

#### **forward\_signal**

If true, trace signal and forward signal from 'in' to 'out'. If false, trace signal only without driving 'out' port.

Type: bool



Default value: `true`

## 3.393 Signal\_Multiplexer

Defined in `LISA/Multiplexer.lisa`.

### About Signal\_Multiplexer

Signal Multiplexer.

### Iris and MTI instances for Signal\_Multiplexer

This model has the following Iris instances:

Name	Instance type
Signal_Multiplexer	Signal_Multiplexer

### Ports for Signal\_Multiplexer

Port	Direction	Protocol	Description
input	slave	Signal	-
output	master	Signal	-
selector	slave	Value	-

### Parameters for Signal\_Multiplexer

#### diagnostics

Diagnostics.

Type: `uint8_t`

Default value: 2

## 3.394 SimplePVBUSMaster

Defined in `LISA/SimplePVBUSMaster.lisa`.

### About SimplePVBUSMaster

Component to generate PVTransactions with configurable attributes and address.

### Iris and MTI instances for SimplePVBUSMaster

This model has the following Iris instances:

Name	Instance type
SimplePVBUSMaster	SimplePVBUSMaster

Name	Instance type
SimplePVBUSMaster.clocktimer64	ClockTimerThread64
SimplePVBUSMaster.clocktimer64.thread	SchedulerThread
SimplePVBUSMaster.clocktimer64.thread_event	SchedulerThreadEvent
SimplePVBUSMaster.pvbusmaster	PVBusMaster
SimplePVBUSMaster.pvbusslave	PVBusSlave

This model has the following MTI trace components:

Name	Component type
SimplePVBUSMaster.pvbusmaster	PVBusMaster
SimplePVBUSMaster.pvbusslave	PVBusSlave

### Ports for SimplePVBUSMaster

Port	Direction	Protocol	Description
pvbus_m	master	PVBus	Output of generated transactions.
pvbus_s	slave	PVBus	-

### Parameters for SimplePVBUSMaster

#### **verbose**

verbose.

Type: `bool`

Default value: `false`

## 3.395 SoC\_SOR

Defined in `LISA/SoC_SOR.lisa`.

### About SoC\_SOR

System Override Registers unit.

### Iris and MTI instances for SoC\_SOR

This model has the following Iris instances:

Name	Instance type
SoC_SOR	SoC_SOR
SoC_SOR.hd1cdX_override (where X = 0-1)	PVBusMapper
SoC_SOR.pvbusslave	PVBusSlave

This model has the following MTI trace components:

Name	Component type
SoC_SOR.hdlcdX_override (where X = 0-1)	PVBusMapper
SoC_SOR.pvbusslave	PVBusSlave

### Ports for SoC\_SOR

Port	Direction	Protocol	Description
hdlcd0_pvbus_s	slave	PVBus	-
hdlcd1_pvbus_s	slave	PVBus	-
hdlcd_pvbus_m	master	PVBus	-
pvbus_s	slave	PVBus	-

### Parameters for SoC\_SOR

#### diagnostics

Diagnostics.

Type: uint32\_t

Default value: 0

#### gpr0

General Purpose Register 0.

Type: uint32\_t

Default value: 0

#### gpr1

General Purpose Register 1.

Type: uint32\_t

Default value: 0

## 3.396 SoC\_Temperature

Defined in LISA/SoC\_Temperature.lisa.

### About SoC\_Temperature

System On-Chip Temperature component which provides averaged out temperature of Application Processors.

### Iris and MTI instances for SoC\_Temperature

This model has the following Iris instances:

Name	Instance type
SoC_Temperature	SoC_Temperature
SoC_Temperature.target	PVBusSlave

This model has the following MTI trace components:

Name	Component type
SoC_Temperature.target	PVBusSlave

### Ports for SoC\_Temperature

Port	Direction	Protocol	Description
pvbus_reg_s	slave	PVBus	-
temperature_in	slave	ValueState	-

### Parameters for SoC\_Temperature

#### diagnostics

Diagnostics.

Type: uint32\_t

Default value: 0

#### num\_cores

Number of AP cores in the system.

Type: uint32\_t

Default value: 1

## 3.397 SwitchedClockControl

Defined in LISA/SwitchedClockControl.lisa.

### About SwitchedClockControl

Clock control allows input selection, rate control and gating.

### Iris and MTI instances for SwitchedClockControl

This model has the following Iris instances:

Name	Instance type
SwitchedClockControl	SwitchedClockControl
SwitchedClockControl.clkDivX (where X = 1-2)	ClockDivider
SwitchedClockControl.clkGate	ClockGate

Name	Instance type
SwitchedClockControl.clkGate.divider	ClockDivider
SwitchedClockControl.clkSelect	ClockSelector
SwitchedClockControl.clkSelect.clkdivX (where X = 0-10)	ClockDivider
SwitchedClockControl.clkSelect.clkdivider	ClockDivider

This model has the following MTI trace components:

Name	Component type
SwitchedClockControl.clkDivX (where X = 1-2)	ClockDivider
SwitchedClockControl.clkGate.divider	ClockDivider
SwitchedClockControl.clkSelect.clkdivX (where X = 0-10)	ClockDivider
SwitchedClockControl.clkSelect.clkdivider	ClockDivider

### Ports for SwitchedClockControl

Port	Direction	Protocol	Description
clk_out	master	ClockSignal	-
clkDivExt	slave	ClockRateControl	-
clkDivSys	slave	ClockRateControl	-
clkEnable	slave	Signal	-
clkSel	slave	Value	-
halt	master	Signal	-
refClk_in	slave	ClockSignal	-
sysClk_in	slave	ClockSignal	-
xClk_in	slave	ClockSignal	-

### Parameters for SwitchedClockControl

#### **clkDiv1.div**

Clock Rate Divider.

Type: uint64\_t

Default value: 1

#### **clkDiv1.mul**

Clock Rate Multiplier.

Type: uint64\_t

Default value: 1

#### **clkDiv2.div**

Clock Rate Divider.

Type: `uint64_t`

Default value: 1

#### **`clkDiv2.mul`**

Clock Rate Multiplier.

Type: `uint64_t`

Default value: 1

#### **`clkGate.diagnostics`**

Diagnostics.

Type: `uint32_t`

Default value: 0

#### **`clkGate.divider.div`**

Clock Rate Divider.

Type: `uint64_t`

Default value: 1

#### **`clkGate.divider.mul`**

Clock Rate Multiplier.

Type: `uint64_t`

Default value: 1

#### **`clkSelect.clkdiv0.div`**

Clock Rate Divider.

Type: `uint64_t`

Default value: 1

#### **`clkSelect.clkdiv0.mul`**

Clock Rate Multiplier.

Type: `uint64_t`

Default value: 1

#### **`clkSelect.clkdiv1.div`**

Clock Rate Divider.

Type: `uint64_t`

Default value: 1

**`clkSelect.clkdiv1.mul`**

Clock Rate Multiplier.

Type: `uint64_t`

Default value: 1

**`clkSelect.clkdiv10.div`**

Clock Rate Divider.

Type: `uint64_t`

Default value: 1

**`clkSelect.clkdiv10.mul`**

Clock Rate Multiplier.

Type: `uint64_t`

Default value: 1

**`clkSelect.clkdiv2.div`**

Clock Rate Divider.

Type: `uint64_t`

Default value: 1

**`clkSelect.clkdiv2.mul`**

Clock Rate Multiplier.

Type: `uint64_t`

Default value: 1

**`clkSelect.clkdiv3.div`**

Clock Rate Divider.

Type: `uint64_t`

Default value: 1

**`clkSelect.clkdiv3.mul`**

Clock Rate Multiplier.

Type: `uint64_t`

Default value: 1

#### **`clkSelect.clkdiv4.div`**

Clock Rate Divider.

Type: `uint64_t`

Default value: 1

#### **`clkSelect.clkdiv4.mul`**

Clock Rate Multiplier.

Type: `uint64_t`

Default value: 1

#### **`clkSelect.clkdiv5.div`**

Clock Rate Divider.

Type: `uint64_t`

Default value: 1

#### **`clkSelect.clkdiv5.mul`**

Clock Rate Multiplier.

Type: `uint64_t`

Default value: 1

#### **`clkSelect.clkdiv6.div`**

Clock Rate Divider.

Type: `uint64_t`

Default value: 1

#### **`clkSelect.clkdiv6.mul`**

Clock Rate Multiplier.

Type: `uint64_t`

Default value: 1

#### **`clkSelect.clkdiv7.div`**

Clock Rate Divider.



Type: `uint64_t`

Default value: 1

#### **`clkSelect.clkdiv7.mul`**

Clock Rate Multiplier.

Type: `uint64_t`

Default value: 1

#### **`clkSelect.clkdiv8.div`**

Clock Rate Divider.

Type: `uint64_t`

Default value: 1

#### **`clkSelect.clkdiv8.mul`**

Clock Rate Multiplier.

Type: `uint64_t`

Default value: 1

#### **`clkSelect.clkdiv9.div`**

Clock Rate Divider.

Type: `uint64_t`

Default value: 1

#### **`clkSelect.clkdiv9.mul`**

Clock Rate Multiplier.

Type: `uint64_t`

Default value: 1

#### **`clkSelect.clkdivider.div`**

Clock Rate Divider.

Type: `uint64_t`

Default value: 1

#### **`clkSelect.clkdivider.mul`**

Clock Rate Multiplier.

Type: uint64\_t

Default value: 1

### **clkSelect.diagnostics**

Diagnostics.

Type: uint32\_t

Default value: 0

## 3.398 SystemC2Clock

Defined in `examples/SystemCExport/Bridges/SystemC2Clock.lisa`.

### About SystemC2Clock

Clock to SystemC Converter.

### Iris and MTI instances for SystemC2Clock

This model has the following Iris instances:

Name	Instance type
SystemC2Clock	SystemC2Clock

### Ports for SystemC2Clock

Port	Direction	Protocol	Description
clk_out	master	ClockSignal	ClockSignal output
current_ticks_m	master	AMBAPVValueState64	To SystemC.
get_clock_m	master	AMBAPVValueState64	To SystemC.
rate_in_Hz_m	master	AMBAPVValueState64	To SystemC.
set_clock_s	slave	AMBAPVValue64	From SystemC.

### Parameters for SystemC2Clock

This component does not have any parameters.

## 3.399 SystemC2Coprocbus

Defined in `examples/SystemCExport/Bridges/SystemC2Coprocbus.lisa`.

### About SystemC2Coprocbus

SystemCCoprocbusProtocol to CoprocbusProtocol converter.

## Iris and MTI instances for SystemC2CprocBus

This model has the following Iris instances:

Name	Instance type
SystemC2CprocBus	SystemC2CprocBus

## Ports for SystemC2CprocBus

Port	Direction	Protocol	Description
coproc_bus_m	master	CoprocBusProtocol	-
sc_coproc_bus_s	slave	SystemCCoprocBusProtocol	-

## Parameters for SystemC2CprocBus

This component does not have any parameters.

# 3.400 SystemC2CounterInterface

Defined in `examples/SystemCExport/Bridges/SystemC2CounterInterface.lisa`.

## About SystemC2CounterInterface

SystemC to CounterInterface Converter.

## Iris and MTI instances for SystemC2CounterInterface

This model has the following Iris instances:

Name	Instance type
SystemC2CounterInterface	SystemC2CounterInterface

## Ports for SystemC2CounterInterface

Port	Direction	Protocol	Description
amba_pv_eventUpdate_s	slave	AMBAPVValue	From SystemC.
amba_pv_getCounterValue_m	master	AMBAPVValueState64	To SystemC.
amba_pv_requestEventUpdate_m	master	AMBAPVValue64	To SystemC.
amba_pv_requestSignalUpdate_m	master	AMBAPVValue64	To SystemC.
amba_pv_setEnabled_s	slave	AMBAPVValue	From SystemC.
amba_pv_signalUpdate_s	slave	AMBAPVValue	From SystemC.
cntvalueb	master	CounterInterface	-

## Parameters for SystemC2CounterInterface

This component does not have any parameters.

## 3.401 SystemC2InstructionCount

Defined in `examples/SystemCExport/Bridges/SystemC2InstructionCount.lisa`.



Note

Variants of this component also exist with multiple input and output ports.

### Iris and MTI instances for SystemC2InstructionCount

This model has the following Iris instances:

Name	Instance type
SystemC2InstructionCount	SystemC2InstructionCount

### Ports for SystemC2InstructionCount

Port	Direction	Protocol	Description
inst_count	master	AMBAPVValueState64	To SystemC to request instruction count.
run_state	master	AMBAPVValueState	To SystemC to request run state.
ticks	master	InstructionCount	InstructionCount input.

### Parameters for SystemC2InstructionCount

This component does not have any parameters.

## 3.402 SystemC2InstructionCountx4

Defined in `examples/SystemCExport/Bridges/SystemC2InstructionCountx4.lisa`.

### About SystemC2InstructionCountx4

SystemC to InstructionCount Coverter x4.

### Iris and MTI instances for SystemC2InstructionCountx4

This model has the following Iris instances:

Name	Instance type
SystemC2InstructionCountx4	SystemC2InstructionCountx4

### Ports for SystemC2InstructionCountx4

Port	Direction	Protocol	Description
inst_count	master	AMBAPVValueState64	-
run_state	master	AMBAPVValueState	-
ticks	master	InstructionCount	-

### Parameters for SystemC2InstructionCountx4

This component does not have any parameters.

## 3.403 SystemC2InstructionCountx8

Defined in `examples/SystemCExport/Bridges/SystemC2InstructionCountx8.lisa`.

### About SystemC2InstructionCountx8

SystemC to InstructionCount Coverter x8.

### Iris and MTI instances for SystemC2InstructionCountx8

This model has the following Iris instances:

Name	Instance type
SystemC2InstructionCountx8	SystemC2InstructionCountx8

### Ports for SystemC2InstructionCountx8

Port	Direction	Protocol	Description
inst_count	master	AMBAPVValueState64	-
run_state	master	AMBAPVValueState	-
ticks	master	InstructionCount	-

### Parameters for SystemC2InstructionCountx8

This component does not have any parameters.

## 3.404 SystemC2LCD

Defined in `examples/SystemCExport/Bridges/SystemC2LCD.lisa`.

### About SystemC2LCD

Converts SystemC to LCD protocol.

### Iris and MTI instances for SystemC2LCD

This model has the following Iris instances:

Name	Instance type
SystemC2LCD	SystemC2LCD

### Ports for SystemC2LCD

Port	Direction	Protocol	Description
all_received_sPL	slave	AMBAPVSignal	From SystemC.

Port	Direction	Protocol	Description
all_received_u	slave	AMBAPVSignal	From SystemC.
lcd_m	master	LCD	LCD output.
lock_s	slave	AMBAPVValueState64	From SystemC.
setPreferredLayout_d	slave	AMBAPVValue	From SystemC.
setPreferredLayout_h	slave	AMBAPVValue	From SystemC.
setPreferredLayout_w	slave	AMBAPVValue	From SystemC.
unlock_s	slave	AMBAPVSignal	From SystemC.
update_h	slave	AMBAPVValue	From SystemC.
update_w	slave	AMBAPVValue	From SystemC.
update_x	slave	AMBAPVValue	From SystemC.
update_y	slave	AMBAPVValue	From SystemC.

### Parameters for SystemC2LCD

This component does not have any parameters.

## 3.405 SystemC2PChannel

Defined in `examples/SystemCExport/Bridges/SystemC2PChannel.lisa`.

### About SystemC2PChannel

SystemC to PChannel Converter.

### Iris and MTI instances for SystemC2PChannel

This model has the following Iris instances:

Name	Instance type
SystemC2PChannel	SystemC2PChannel

### Ports for SystemC2PChannel

Port	Direction	Protocol	Description
pchannel	master	PChannel	-
sc_pchannel	slave	SystemCPChannel	-

### Parameters for SystemC2PChannel

This component does not have any parameters.

## 3.406 SystemC2VirtualEthernet

Defined in `examples/SystemCExport/Bridges/SystemC2VirtualEthernet.lisa`.

### About SystemC2VirtualEthernet

SystemC to VirtualEthernet Converter.

### Iris and MTI instances for SystemC2VirtualEthernet

This model has the following Iris instances:

Name	Instance type
SystemC2VirtualEthernet	SystemC2VirtualEthernet

### Ports for SystemC2VirtualEthernet

Port	Direction	Protocol	Description
virtualethernet_m	master	VirtualEthernet	-
virtualethernet_s	slave	SC_VirtualEthernet	-

### Parameters for SystemC2VirtualEthernet

This component does not have any parameters.

## 3.407 SystemC2v7VGICConfig

Defined in `examples/SystemCExport/Bridges/SystemC2v7VGICConfig.lisa`.

### About SystemC2v7VGICConfig

Converts SystemC to v7\_vgic\_configuration\_protocol.

### Iris and MTI instances for SystemC2v7VGICConfig

This model has the following Iris instances:

Name	Instance type
SystemC2v7VGICConfig	SystemC2v7VGICConfig

### Ports for SystemC2v7VGICConfig

Port	Direction	Protocol	Description
all_received_s	slave	AMBAPVSignal	From SystemC.
cpu_interface_number_s	slave	AMBAPVValue64	From SystemC.
inout_cluster_number_s	slave	AMBAPVValue64	From SystemC.
inout_cpu_number_in_cluster_s	slave	AMBAPVValue64	From SystemC.
manager_id_mask_s	slave	AMBAPVValue64	From SystemC.
manager_id_s	slave	AMBAPVValue64	From SystemC.

Port	Direction	Protocol	Description
number_of_cores_s	slave	<a href="#">AMBAPVValueState</a>	From SystemC.
response_m	master	<a href="#">AMBAPVSignal</a>	To SystemC.
v7_vgic_config_m	master	<a href="#">v7_VGIC_Configuration_Protocol</a>	v7_VGIC_Configuration_Protocol output.

### Parameters for SystemC2v7VGICConfig

This component does not have any parameters.

## 3.408 SystemFMU

Defined in `LISA/SystemFMU.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### About SystemFMU

System level Fault Management Unit.

### Iris and MTI instances for SystemFMU

This model has the following Iris instances:

Name	Instance type
SystemFMU	<a href="#">SYSTEM_FMU</a>
SystemFMU.pvbus_slave	<a href="#">PVBusSlave</a>

This model has the following MTI trace components:

Name	Component type
SystemFMU	<a href="#">SYSTEM_FMU</a>
SystemFMU.pvbus_slave	<a href="#">PVBusSlave</a>

### Ports for SystemFMU

Port	Direction	Protocol	Description
c_error_in	slave	<a href="#">Signal</a>	Critical interrupts from the device side FMUs
c_error_out	master	<a href="#">Signal</a>	Critical error input from the system FMU
clk_in	slave	<a href="#">ClockSignal</a>	Clock input
cold_reset_in	slave	<a href="#">Signal</a>	Cold reset signal
nc_error_in	slave	<a href="#">Signal</a>	Non critical interrupts from the device side FMUs
nc_error_out	master	<a href="#">Signal</a>	Non critical error input from the system FMU



Port	Direction	Protocol	Description
pvbuss_s	slave	PVBus	To access FMU model registers
warm_reset_in	slave	Signal	Warm reset signal

## Parameters for SystemFMU

### **diagnostics**

Diagnostics.

Type: uint8\_t

Default value: 2

### **upstream\_fmuscfg**

Upstream FMUs which are to be connected to System FMU.

Type: uint8\_t

Default value: 1

## 3.409 SystemIdUnit

Defined in LISA/SystemIdUnit.lisa.

### About SystemIdUnit

System ID Unit.

### Iris and MTI instances for SystemIdUnit

This model has the following Iris instances:

Name	Instance type
SystemIdUnit	SystemIdUnit
SystemIdUnit.pvbusslave	PVBusSlave

This model has the following MTI trace components:

Name	Component type
SystemIdUnit.pvbusslave	PVBusSlave

### Ports for SystemIdUnit

Port	Direction	Protocol	Description
pvbuss_s	slave	PVBus	-

## Parameters for SystemIdUnit

### **chip\_id**

the ID for the node/chip when there are multiple SoCs.

Type: `uint8_t`

Default value: `0`

### **chiplet\_type**

Specifies Compute Chiplet (CC) or Specialization Chiplet (SC). 0 - CC, 1 - SC.

Type: `bool`

Default value: `false`

### **diagnostics**

Diagnostics.

Type: `uint32_t`

Default value: `2`

### **multi\_chip\_mode**

Multi chip mode?.

Type: `bool`

Default value: `false`

### **platform\_type**

the type of the subsystem: 0=mobile, 1=InfraSysDesign4.x, 2=InfraSysDesign5.x, 3=InfraSysDesign6.x, 4=RD1AE SafetyIsland, 5=Client, 6=RDASD.

Type: `uint32_t`

Default value: `0`

### **soc\_id**

the ID for the SoC that integrates the subsystem.

Type: `uint32_t`

Default value: `0`

### **system\_cfg**

the ID for the subsystem configuration.

Type: uint32\_t

Default value: 0

### **system\_id**

the version ID for the subsystem.

Type: uint32\_t

Default value: 0

## 3.410 System\_RAS\_Agent

Defined in LISA/System\_RAS\_Agent.lisa.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### About System\_RAS\_Agent

System level RAS agent.

### Iris and MTI instances for System\_RAS\_Agent

This model has the following Iris instances:

Name	Instance type
System_RAS_Agent	SYSTEM_RAS_AGENT
System_RAS_Agent.pvbus_subordinate	PVBusSlave

This model has the following MTI trace components:

Name	Component type
System_RAS_Agent.pvbus_subordinate	PVBusSlave

### Ports for System\_RAS\_Agent

Port	Direction	Protocol	Description
apb	slave	PVBus	To access RAS agent registers
clk_in	slave	ClockSignal	Input Clock - RAS agent is in this clock domain
cri_in	slave	Signal	Critical Error Interrupt from the downstream RAS agent
cri_out	master	Signal	CRI_OUT is the consolidated status of the Critical Error Interrupt from this and the downstream RAS agent(s).

Port	Direction	Protocol	Description
eri_in	slave	Signal	Error Recovery Interrupt from the downstream RAS agent
eri_out	master	Signal	consolidated status of the Error Recovery Interrupt from this and the downstream RAS agent(s).
fhi_in	slave	Signal	Fault Handling Interrupt from the downstream RAS agent
fhi_out	master	Signal	consolidated status of the Fault Handling Interrupt from this and the downstream RAS agent(s).
reset_in	slave	Signal	Input reset - Connect to the system cold reset.
valid_in	slave	Signal	Incoming Valid from a downstream RAS agent indicates the presence of at least one valid error record.
valid_out	master	Signal	outgoing Valid from the downstream RAS agent(s) contains at least one valid error record.

## Parameters for System\_RAS\_Agent

### NUM\_DOWNSTREAM\_RAS\_AGENTS

Number of downstream RAS agents for which the proxy error record is maintained.

Type: uint8\_t

Default value: 32

### SYNC\_ENABLE

Enables synchronization on the Interrupt lines and valid line before assigning it to the corresponding bits in the ERR<n>STATUS register. 1-bit per downstream RAS Agent.

Type: uint64\_t

Default value: 0x0

### diagnostics

Diagnostics.

Type: uint8\_t

Default value: 2

## 3.411 TRNG

Defined in LISA/TRNG.lisa.

### About TRNG

True Random Number Generator.

### Iris and MTI instances for TRNG

This model has the following Iris instances:

Name	Instance type
TRNG	TRNG
TRNG.pvbusslave	PVBusSlave

This model has the following MTI trace components:

Name	Component type
TRNG.pvbusslave	PVBusSlave

### Ports for TRNG

Port	Direction	Protocol	Description
cc_host_int_req	master	Signal	-
pvbus_s	slave	PVBus	-
rng_clk	slave	ClockSignal	-
rst_n	slave	Signal	-
scanmode	slave	Signal	-

### Parameters for TRNG

#### diagnostics

Diagnostics.

Type: uint32\_t

Default value: 0

## 3.412 TZC\_400

Defined in LISA/TZC\_400.lisa.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### About TZC\_400

The TZC-400 determines, under software control, whether a particular bus master is permitted to issue Non-secure accesses to a particular physical address.

The component has:

- Eight address regions in addition to the base region, region 0.

- A programmable control block for security-access permissions configuration through the Advanced Peripheral Bus (APB).
- Up to four address filters that share common set region set-up registers.
- Software configurable permission check failure reporting and interrupt signaling.
- Filtering with a Non-Secure Access ID (NSAID).
- A gate keeper, to allow or block accesses to the filter unit.
- Configurable reset values of region configuration registers and other key configuration registers.

This component has the following subcomponents:

### TZFilterUnits

The TZC-400 has four TZFilterUnits. The `BUILD_CONFIG` register sets the configuration. The `rst_build_config` parameter controls the register. The value of `rst_build_config` varies with the system. See the system design documentation or system integration documentation. For AEMvA, it is `0x3003F08`.

### TZDummyDevice

An internal dummy device that mimics **RAZ/WI** for TZFilterUnits. The system uses it when there is a permission violation and a bus returns Transaction OK.

### Differences between the model and the RTL

Unlike the hardware, this component does not have:

- Asynchronous clocks. The model does not need clocks for data transfer, or clock signals.
- QoS Virtual Network (QVN) support. Specifically, it does not implement the `vnet` bits[27:24] in `FAIL_ID_<x>` registers.
- Fast Path and Fast Path ID. In the model, transactions occur at similar speeds.
- 256 outstanding accesses globally for each read or write Normal Paths and configurable 8, 16, or 32 outstanding accesses on Fast Path read access. The model does not support QVN. This concept is meaningless for a PV level model.
- Configurable address bus width, data bus width, transaction ID tag, and USER bus width. A single bus implementation, PVBUS, covers these AXI bus hardware implementation details.

This component does not implement:

- The `vnet` bits[27:24] in `FAIL_ID_<x>` registers.
- Any background logic for the speculation control register. This does not affect model behavior.

### Configuration

- Configure `manager_id_from_label` OR `id_mapping`, `rst_build_config`, and `rst_region_attributes_0` before running the model to set the desired behaviors. Otherwise, the system resets all region configuration registers, `rst_action`, and `rst_gate_keeper` to 0, and resets `rst_build_config` and `rst_region_attributes_0` to sensible default values.
- Configure either `id_mapping` OR `manager_id_from_label` at model init, or a warning message appears.

- The syntax of `id_mapping` is:

```
<managerid_0>:<nsaid_0>,<managerid_1>:<nsaid_1>,<managerid_n>:<nsaid_n>
```

Separate the mapping pairs by a comma. The `managerid` is the ID of the bus master, such as the parameter `CLUSTER_ID` on Cortex-A15/7, `cluster_id` port of Cortex-A15/7, or `manager_id` parameter for Cortex-M3.

## Iris and MTI instances for TZC\_400

This model has the following Iris instances:

Name	Instance type
TZC_400	TZC_400
TZC_400.ExportTest.TZC_400[Z].pvbusmaster (where Z = 0-3)	PVBusMaster
TZC_400.apbslave[0]	PVBusSlave
TZC_400.filter0	filter0
TZC_400.filterY.BusMapper (where Y = 0-3)	PVBusMapper
TZC_400.filter1	filter1
TZC_400.filter2	filter2
TZC_400.filter3	filter3

This model has the following MTI trace components:

Name	Component type
TZC_400.ExportTest.TZC_400[Z].pvbusmaster (where Z = 0-3)	PVBusMaster
TZC_400.apbslave[0]	PVBusSlave
TZC_400.filterY.BusMapper (where Y = 0-3)	PVBusMapper

## Ports for TZC\_400

Port	Direction	Protocol	Description
apbslave_s	slave	PVBus	Bus access for control register.
filter_pvbus_m	master	PVBus	Outgoing bus traffic from filter units.
filter_pvbus_s	slave	PVBus	Incoming bus traffic to filter units.
tzc_reset	slave	Signal	Reset signal from external master.
tzcint	master	Signal	TrustZone interrupt signal, controlled by ACTION register.

## Parameters for TZC\_400

### diagnostics

Diagnostics.

Type: `uint32_t`

Default value: 0

**id\_mapping**

Mapping from Manager ID to NSAID.

Type: `string`

Default value: `"0:0,1:0,2:0,3:0,4:0,5:0,6:0,7:0,8:0,9:0,10:0,11:0,12:0,13:0,14:0,15:0"`

**manager\_id\_from\_label**

Obtain Manager ID from label (ignores `id_mapping`).

Type: `bool`

Default value: `false`

**rst\_action**

ACTION register value at reset.

Type: `uint32_t`

Default value: 0

**rst\_build\_config**

BUILD\_CONFIG register value at reset.

Type: `uint32_t`

Default value: `0x03003F08`

**rst\_gate\_keeper**

GATE\_KEEPER register value at reset.

Type: `uint32_t`

Default value: 0

**rst\_region\_attributes\_0**

Region 0 Secure attributes.

Type: `uint32_t`

Default value: `0x0f`

**rst\_region\_attributes\_1**

Region 1 Secure attributes.

Type: `uint32_t`

Default value: 0



**rst\_region\_attributes\_2**

Region 2 Secure attributes.

Type: uint32\_t

Default value: 0

**rst\_region\_attributes\_3**

Region 3 Secure attributes.

Type: uint32\_t

Default value: 0

**rst\_region\_attributes\_4**

Region 4 Secure attributes.

Type: uint32\_t

Default value: 0

**rst\_region\_attributes\_5**

Region 5 Secure attributes.

Type: uint32\_t

Default value: 0

**rst\_region\_attributes\_6**

Region 6 Secure attributes.

Type: uint32\_t

Default value: 0

**rst\_region\_attributes\_7**

Region 7 Secure attributes.

Type: uint32\_t

Default value: 0

**rst\_region\_attributes\_8**

Region 8 Secure attributes.

Type: uint32\_t

Default value: 0

**rst\_region\_base\_high\_1**

Region 1 base memory address (high 32 bits).

Type: uint32\_t

Default value: 0

**rst\_region\_base\_high\_2**

Region 2 base memory address (high 32 bits).

Type: uint32\_t

Default value: 0

**rst\_region\_base\_high\_3**

Region 3 base memory address (high 32 bits).

Type: uint32\_t

Default value: 0

**rst\_region\_base\_high\_4**

Region 4 base memory address (high 32 bits).

Type: uint32\_t

Default value: 0

**rst\_region\_base\_high\_5**

Region 5 base memory address (high 32 bits).

Type: uint32\_t

Default value: 0

**rst\_region\_base\_high\_6**

Region 6 base memory address (high 32 bits).

Type: uint32\_t

Default value: 0

**rst\_region\_base\_high\_7**

Region 7 base memory address (high 32 bits).

Type: uint32\_t

Default value: 0

**rst\_region\_base\_high\_8**

Region 8 base memory address (high 32 bits).

Type: uint32\_t

Default value: 0

**rst\_region\_base\_low\_1**

Region 1 base memory address (low 32 bits).

Type: uint32\_t

Default value: 0

**rst\_region\_base\_low\_2**

Region 2 base memory address (low 32 bits).

Type: uint32\_t

Default value: 0

**rst\_region\_base\_low\_3**

Region 3 base memory address (low 32 bits).

Type: uint32\_t

Default value: 0

**rst\_region\_base\_low\_4**

Region 4 base memory address (low 32 bits).

Type: uint32\_t

Default value: 0

**rst\_region\_base\_low\_5**

Region 5 base memory address (low 32 bits).

Type: uint32\_t

Default value: 0

**rst\_region\_base\_low\_6**

Region 6 base memory address (low 32 bits).

Type: uint32\_t

Default value: 0

**rst\_region\_base\_low\_7**

Region 7 base memory address (low 32 bits).

Type: uint32\_t

Default value: 0

**rst\_region\_base\_low\_8**

Region 8 base memory address (low 32 bits).

Type: uint32\_t

Default value: 0

**rst\_region\_id\_access\_0**

Region 0 NSAID permissions.

Type: uint32\_t

Default value: 0

**rst\_region\_id\_access\_1**

Region 1 NSAID permissions.

Type: uint32\_t

Default value: 0

**rst\_region\_id\_access\_2**

Region 2 NSAID permissions.

Type: uint32\_t

Default value: 0

**rst\_region\_id\_access\_3**

Region 3 NSAID permissions.

Type: uint32\_t

Default value: 0

**rst\_region\_id\_access\_4**

Region 4 NSAID permissions.

Type: uint32\_t

Default value: 0

**rst\_region\_id\_access\_5**

Region 5 NSAID permissions.

Type: uint32\_t

Default value: 0

**rst\_region\_id\_access\_6**

Region 6 NSAID permissions.

Type: uint32\_t

Default value: 0

**rst\_region\_id\_access\_7**

Region 7 NSAID permissions.

Type: uint32\_t

Default value: 0

**rst\_region\_id\_access\_8**

Region 8 NSAID permissions.

Type: uint32\_t

Default value: 0

**rst\_region\_top\_high\_0**

Region 0 (default) top memory address.

Type: uint32\_t

Default value: 0

**rst\_region\_top\_high\_1**

Region 1 top memory address (high 32 bits).

Type: uint32\_t

Default value: 0

**rst\_region\_top\_high\_2**

Region 2 top memory address (high 32 bits).

Type: uint32\_t

Default value: 0

**rst\_region\_top\_high\_3**

Region 3 top memory address (high 32 bits).

Type: uint32\_t

Default value: 0

**rst\_region\_top\_high\_4**

Region 4 top memory address (high 32 bits).

Type: uint32\_t

Default value: 0

**rst\_region\_top\_high\_5**

Region 5 top memory address (high 32 bits).

Type: uint32\_t

Default value: 0

**rst\_region\_top\_high\_6**

Region 6 top memory address (high 32 bits).

Type: uint32\_t

Default value: 0

**rst\_region\_top\_high\_7**

Region 7 top memory address (high 32 bits).

Type: uint32\_t

Default value: 0

**rst\_region\_top\_high\_8**

Region 8 top memory address (high 32 bits).

Type: uint32\_t

Default value: 0

**rst\_region\_top\_low\_1**

Region 1 top memory address (low 32 bits).

Type: uint32\_t

Default value: 0

**rst\_region\_top\_low\_2**

Region 2 top memory address (low 32 bits).

Type: uint32\_t

Default value: 0

**rst\_region\_top\_low\_3**

Region 3 top memory address (low 32 bits).

Type: uint32\_t

Default value: 0

**rst\_region\_top\_low\_4**

Region 4 top memory address (low 32 bits).

Type: uint32\_t

Default value: 0

**rst\_region\_top\_low\_5**

Region 5 top memory address (low 32 bits).

Type: uint32\_t

Default value: 0

**rst\_region\_top\_low\_6**

Region 6 top memory address (low 32 bits).

Type: uint32\_t

Default value: 0

**rst\_region\_top\_low\_7**

Region 7 top memory address (low 32 bits).

Type: uint32\_t

Default value: 0

**rst\_region\_top\_low\_8**

Region 8 top memory address (low 32 bits).

Type: uint32\_t

Default value: 0

## 3.413 TZFilterUnit

Defined in `LISA/TZFilterUnit.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### About TZFilterUnit

TrustZone Filter Unit.

### Iris and MTI instances for TZFilterUnit

This model has the following Iris instances:

Name	Instance type
<code>TZFilterUnit</code>	<a href="#">TZFilterUnit</a>
<code>TZFilterUnit.BusMapper</code>	<a href="#">PVBusMapper</a>

This model has the following MTI trace components:

Name	Component type
<code>TZFilterUnit.BusMapper</code>	<a href="#">PVBusMapper</a>

### Ports for TZFilterUnit

Port	Direction	Protocol	Description
<code>control</code>	master	<a href="#">TZFilterControl</a>	Configuration port.
<code>pvbus_m</code>	master	<a href="#">PVBus</a>	Master bus port.
<code>pvbus_s</code>	slave	<a href="#">PVBus</a>	Slave bus port.

### Parameters for TZFilterUnit

This component does not have any parameters.

## 3.414 TZIC

Defined in `LISA/TZIC.lisa`.

This model supports the following revisions of the IP at the given quality levels:



Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

## About TZIC

The TZIC provides a software interface to the secure interrupt system in a TrustZone design. It provides secure control of the nFIQ and masks out the interrupt sources chosen for nFIQ from the interrupts that are passed onto a non-secure interrupt controller.

## Iris and MTI instances for TZIC

This model has the following Iris instances:

Name	Instance type
TZIC	TZIC
TZIC.busslave	PVBusSlave

This model has the following MTI trace components:

Name	Component type
TZIC.busslave	PVBusSlave

## Ports for TZIC

Port	Direction	Protocol	Description
fiq_out	master	Signal	FIQ interrupt to processor.
input	slave	Signal	32 interrupt input sources.
irq_out	master	Signal	32 IRQ output ports.
nsfiq_in	slave	Signal	Connects to the nFIQ output of the non-secure interrupt controller.
pvbus	slave	PVBus	Slave port for connection to PV bus master/decoder.
sfiq_in	slave	Signal	Daisy chaining secure FIQ input, otherwise connects to logic 1 if interrupt controller not daisy chained.

## Parameters for TZIC

This component does not have any parameters.

# 3.415 TZSwitch

Defined in `LISA/TZSwitch.lisa`.

## About TZSwitch

Transactions received on the `pvbus_input` slave port are routed according to a configuration that is set up using parameters and/or the control port. Separate rules can be given for secure and for normal transactions.

Transactions can be routed to one of the two master ports, `pvbus_port_a` or `pvbus_port_b`, can be ignored, or can generate aborts.

The control port behaviour `routeAccesses()` takes two arguments:

- `input` selects which types of signals are reconfigured:
  - TZINPUT\_SECURE**  
Change the routing for secure transactions
  - TZINPUT\_NORMAL**  
Change the routing for normal transactions
  - TZINPUT\_ANY**  
Change the routing for all transactions
- `destination` selects how the chosen transactions are routed:
  - TZROUTE\_IGNORE**  
Transactions are ignored. Reads return 0.
  - TZROUTE\_TO\_PORT\_A**  
Route transactions to `pvbus_port_a`.
  - TZROUTE\_TO\_PORT\_B**  
Route transactions to `pvbus_port_b`.
  - TZROUTE\_ABORT**  
Cause transactions to generate an abort.

Initial routing is configured using parameters `secure` and `normal` based on the following values:

- 0**  
Ignore these transactions.
- 1**  
Forward the transactions to `pvbus_port_a`.
- 2**  
Forward the transactions to `pvbus_port_b`.
- 3**  
Generate an abort for these transactions.

Both default and explicit parameter values are overridden by any runtime calls to `routeAccesses()` on the control port.

### Iris and MTI instances for TZSwitch

This model has the following Iris instances:

Name	Instance type
TZSwitch	TZSwitch
TZSwitch.pvbus_mapper	PVBusMapper

This model has the following MTI trace components:

Name	Component type
TZSwitch.pvbus_mapper	PVBusMapper

### Ports for TZSwitch

Port	Direction	Protocol	Description
control	slave	TZSwitchControl	Controls routing of transactions.
pvbus_input	slave	PVBus	Slave port for connection to PVBus master/decoder.
pvbus_port_a	master	PVBus	Output port a.
pvbus_port_b	master	PVBus	Output port b.

### Parameters for TZSwitch

#### **normal**

Normal Port.

Type: uint32\_t

Default value: 2

#### **secure**

Secure Port.

Type: uint32\_t

Default value: 1

## 3.416 TelnetTerminal

Defined in LISA/TelnetTerminal.lisa.

### About TelnetTerminal

A host interface onto a serial port: exposes the two way serial data channel over a TCP/IP interface, and automatically opens a telnet application connected to the network socket, unless a user application connects first.

### Iris and MTI instances for TelnetTerminal

This model has the following Iris instances:

Name	Instance type
TelnetTerminal	TelnetTerminal

## Ports for TelnetTerminal

Port	Direction	Protocol	Description
serial	slave	SerialData	Slave port for connecting to a SerialData master.

## Parameters for TelnetTerminal

### **display\_name**

Terminal display name.

Type: `string`

Default value: `""`

### **mode**

Terminal initialisation mode.

Type: `string`

Default value: `"telnet"`

### **quiet**

Avoid output on stdout/stderr.

Type: `bool`

Default value: `"false"`

### **start\_port**

Telnet TCP start port.

Type: `uint32_t`

Default value: `5000`

### **start\_telnet**

Start telnet no matter there is connection or not.

Type: `bool`

Default value: `"true"`

### **terminal\_command**

Commandline to launch a terminal application and connect to the opened TCP port. Keywords `%port` and `%title` will be replaced with the opened port number and component name respectively. An empty string (default behaviour) will launch `xterm` (Linux) or `telnet.exe` (Windows).

Type: `string`

Default value: ""

## 3.417 Temperature

Defined in `LISA/Temperature.lisa`.

### About Temperature

Component to synthesis the temperature value of the connected core.

### Iris and MTI instances for Temperature

This model has the following Iris instances:

Name	Instance type
Temperature	Temperature

### Ports for Temperature

Port	Direction	Protocol	Description
cluster_powerdown	slave	Signal	-
core_powerdown	slave	Signal	-
freq_changed	slave	ValueState	-
temperature	master	ValueState	-

### Parameters for Temperature

#### **core\_coefficient**

Temperature Coefficient.

Type: `uint32_t`

Default value: 0

#### **diagnostics**

Diagnostics.

Type: `uint32_t`

Default value: 0

#### **num\_cores**

Number of cores.

Type: `uint32_t`

Default value: 4

### 3.418 TestbedGPIOConnector

Defined in `LISA/TestbedGPIOConnector.lisa`.

#### About TestbedGPIOConnector

Tool for receiving GPIO signals and reporting test success/failure.

#### Iris and MTI instances for TestbedGPIOConnector

This model has the following Iris instances:

Name	Instance type
TestbedGPIOConnector	TestbedGPIOConnector

#### Ports for TestbedGPIOConnector

Port	Direction	Protocol	Description
from_gpio0	slave	Value	-
from_gpio1	slave	Value	-

#### Parameters for TestbedGPIOConnector

##### active

Actually stop the simulator when testbench running in it reports results over GPIO.

Type: `bool`

Default value: `true`

##### diagnostics

Diagnostic level.

Type: `int`

Default value: `0`

### 3.419 TrustedRAM

Defined in `LISA/TrustedRAM.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
0.4	Alpha support

For an explanation of the quality levels, see [Quality level definitions](#).

## Changes in 11.31.15

The following ports were added:

- key\_changed\_out

## About TrustedRAM

Trusted RAM.

## Iris and MTI instances for TrustedRAM

This model has the following Iris instances:

Name	Instance type
TrustedRAM	<a href="#">TrustedRAM</a>
TrustedRAM.apb	<a href="#">PVBUSlave</a>

This model has the following MTI trace components:

Name	Component type
TrustedRAM.apb	<a href="#">PVBUSlave</a>

## Ports for TrustedRAM

Port	Direction	Protocol	Description
apb	slave	<a href="#">PVBUS</a>	APB Subordinate Interface - Access to registers
key_changed_out	master	<a href="#">Signal</a>	Output to CPU to invalidate ECC tags.
reset_in	slave	<a href="#">Signal</a>	Reset in signal

## Parameters for TrustedRAM

### TRBC\_RESET\_VALUE

TRBC register reset value.

Type: uint32\_t

Default value: 0xB

### diagnostics

Diagnostics.

Type: uint32\_t

Default value: 2

## 3.420 UART\_MUX

Defined in `LISA/UART_MUX.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
1	Alpha support

For an explanation of the quality levels, see [Quality level definitions](#).

### About UART\_MUX

UART MUX Component.

### Iris and MTI instances for UART\_MUX

This model has the following Iris instances:

Name	Instance type
UART_MUX	UART_MUX

### Ports for UART\_MUX

Port	Direction	Protocol	Description
pvbus_reg_s	slave	PVBus	APB Subordinate Port for Register Access
reset_in	slave	Signal	reset port
serial_data_in	slave	SerialData	SerialData Subordinate Ports
serial_data_out_null	master	SerialData	SerialData Manager Ports for NULL
serial_data_out	master	SerialData	SerialData Manager Ports

### Parameters for UART\_MUX

#### diagnostics

Diagnostics.

Type: `uint32_t`

Default value: 2

## 3.421 Ufs

Defined in `LISA/Ufs.lisa`.

### About Ufs

Component for Universal Flash Storage based on JESD220G and JESD223.



## Iris and MTI instances for Ufs

This model has the following Iris instances:

Name	Instance type
Ufs	Ufs
Ufs.doorbell_timer	ClockTimerThread64
Ufs.doorbell_timer.thread	SchedulerThread
Ufs.doorbell_timer.thread_event	SchedulerThreadEvent

## Ports for Ufs

Port	Direction	Protocol	Description
clk_in	slave	ClockSignal	-
pvbus_m	master	PVBus	-
pvbus_s	slave	PVBus	-

## Parameters for Ufs

### NUTMRS

Number of UTP task management request slots.

Type: uint8\_t

Default value: 8

### NUTRS

Number of UTP transfer request slots.

Type: uint8\_t

Default value: 32

### lunConfig

LUN configuration, as a list of pairs of logical block count and logical block sizes (in bytes), separated by spaces like: LBC0:LBS0 LBC1:LBS1 ...

Type: string

Default value: "1:4096"

## 3.422 UnusedPrimeCell

Defined in `LISA/UnusedPrimeCell.lisa`.

### About UnusedPrimeCell

A dummy component. It can be used to represent any unimplemented PrimeCell components.

### Iris and MTI instances for UnusedPrimeCell

This model has the following Iris instances:

Name	Instance type
<code>UnusedPrimeCell</code>	<a href="#">UnusedPrimeCell</a>
<code>UnusedPrimeCell.busslave</code>	<a href="#">PVBusSlave</a>

This model has the following MTI trace components:

Name	Component type
<code>UnusedPrimeCell.busslave</code>	<a href="#">PVBusSlave</a>

### Ports for UnusedPrimeCell

Port	Direction	Protocol	Description
<code>pvbus</code>	slave	<a href="#">PVBus</a>	Bus slave interface.

### Parameters for UnusedPrimeCell

This component does not have any parameters.

## 3.423 V61

Defined in `LISA/V61.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
<code>r0p1</code>	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### About V61

The model requires an external OpenMAX (OMX) IL implementation for codec functionality. FFomaxIL is an OMX IL implementation provided for convenience. By default, V61 looks for `ffomaxil.dll` on Windows or `libffomaxil.so` on Linux in the model binary's directory or in the Fast Models installation. The default library path can be overridden using the `omx-library-path` parameter.

When querying the OMX core, V61 searches for the following roles in the list of OpenMAX components:

#### H.264 decode

"video\_decoder.avc"

#### JPEG decode

"video\_decoder.mjpeg"

#### MPEG2 decode

"video\_decoder.mpeg2"

#### MPEG4 decode

"video\_decoder.mpeg4"

#### VC1 decode

"video\_decoder.vc1"

#### VP8 decode

"video\_decoder.vp8"

#### VP8 encode

"video\_encoder.vp8"

### Limitations

- No support for HEVC, VP9 and RealVideo decoders.
- No support for 10-bit video output.
- No support for RGB or AFBC input for encoding.
- No profiling support.
- No QoS support.
- Power/Test modes are modeled only as register state changes.

### Iris and MTI instances for V61

This model has the following Iris instances:

Name	Instance type
V61	V61
V61.BusModifier.LSIDY (where Y = 0-3)	PVBusMapper
V61.apb_slave[0]	PVBusSlave

This model has the following MTI trace components:

Name	Component type
V61	V61
V61.BusModifier.LSIDY (where Y = 0-3)	PVBusMapper
V61.apb_slave[0]	PVBusSlave

## Ports for V61

Port	Direction	Protocol	Description
apb_s	slave	PVBus	APB Slave port for register access.
axi_m	master	PVBus	AXI master bus for memory accesses.
clk	slave	ClockSignal	Master clock, typically 300MHz.
irq	master	Signal	IRQ signal to host CPU.
reset	slave	Signal	Reset signal.

## Parameters for V61

### **AXI-data-width**

AXI data width, logarithmic byte notation (3->64bit, 4->128bit).

Type: `uint32_t`

Default value: `4`

### **enable-frame-rate-limiting**

Enable output rate control via the CLK port.

Type: `bool`

Default value: `false`

### **fuse-disable-AFBC**

AFBC support disabled by fuse.

Type: `bool`

Default value: `false`

### **fuse-disable-HEVC**

HEVC support disabled by fuse.

Type: `bool`

Default value: `false`

### **fuse-disable-Real**

RealVideo support disabled by fuse.

Type: `bool`

Default value: `false`

**`fuse-disable-VPX`**

VPX support disabled by fuse.

Type: `bool`

Default value: `false`

**`ncores`**

Number of cores in the component.

Type: `uint32_t`

Default value: `1`

**`omx-library-path`**

Path to a user-provided OMX library; leave blank to use FFomaxIL.

Type: `string`

Default value: `""`

**`supports-10bit`**

Component supports 10-bit content decoding.

Type: `bool`

Default value: `true`

**`supports-64byte-ref-bursts`**

Component supports 64-byte bursts for reference pixel data.

Type: `bool`

Default value: `true`

**`supports-encoding`**

Component supports encoding.

Type: `bool`

Default value: `true`

## 3.424 V76

Defined in `LISA/V76.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r0p1	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### About V76

The model requires an external OpenMAX (OMX) IL implementation for codec functionality. FFomaxIL is an OMX IL implementation provided for convenience. By default, V76 looks for `ffomaxil.dll` on Windows or `libffomaxil.so` on Linux in the model binary's directory or in the Fast Models installation. The default library path can be overridden using the `omx-library-path` parameter.

When querying the OMX core, V76 searches for the following roles in the list of OpenMAX components:

#### H.264 decode

"video\_decoder.avc"

#### JPEG decode

"video\_decoder.mjpeg"

#### MPEG2 decode

"video\_decoder.mpeg2"

#### MPEG4 decode

"video\_decoder.mpeg4"

#### VC1 decode

"video\_decoder.vc1"

#### VP8 decode

"video\_decoder.vp8"

#### VP8 encode

"video\_encoder.vp8"

### Limitations

- No support for HEVC, VP9 and RealVideo decoders.
- No support for 10-bit video output.
- No support for RGB or AFBC input for encoding.
- No profiling support.
- No QoS support.

- Power/Test modes are modeled only as register state changes.

## Iris and MTI instances for V76

This model has the following Iris instances:

Name	Instance type
V76	V76
V76.BusModifier.LSIDY (where Y = 0–3)	PVBusMapper
V76.apb_slave[0]	PVBusSlave

This model has the following MTI trace components:

Name	Component type
V76	V76
V76.BusModifier.LSIDY (where Y = 0–3)	PVBusMapper
V76.apb_slave[0]	PVBusSlave

## Ports for V76

Port	Direction	Protocol	Description
apb_s	slave	PVBus	APB Slave port for register access.
axi_m	master	PVBus	AXI master bus for memory accesses.
clk	slave	ClockSignal	Master clock, typically 300MHz.
irq	master	Signal	IRQ signal to host CPU.
reset	slave	Signal	Reset signal.

## Parameters for V76

### **enable-frame-rate-limiting**

Enable output rate control via the CLK port.

Type: `bool`

Default value: `false`

### **fuse-disable-AFBC**

AFBC support disabled by fuse.

Type: `bool`

Default value: `false`

### **fuse-disable-HEVC**

HEVC support disabled by fuse.

Type: `bool`

Default value: `false`

**`fuse-disable-Real`**

RealVideo support disabled by fuse.

Type: `bool`

Default value: `false`

**`fuse-disable-VPX`**

VPX support disabled by fuse.

Type: `bool`

Default value: `false`

**`ncores`**

Number of cores in the component.

Type: `uint32_t`

Default value: `2`

**`omx-library-path`**

Path to a user-provided OMX library; leave blank to use FFomaxIL.

Type: `string`

Default value: `""`

**`supports-10bit`**

Component supports 10-bit content decoding.

Type: `bool`

Default value: `true`

**`supports-64byte-ref-bursts`**

Component supports 64-byte bursts for reference pixel data.

Type: `bool`

Default value: `true`

**`supports-8k`**

Component supports 8K content.

Type: `bool`



Default value: `true`

**supports-encoding**

Component supports encoding.

Type: `bool`

Default value: `true`

3.425 VECB2AMBAPVValue64

Defined in `examples/SystemCEExport/Bridges/VECB2AMBAPVValue64.lisa`.

**About VECB2AMBAPVValue64**

VECB protocol to AMBA-PV protocol converter.

**Iris and MTI instances for VECB2AMBAPVValue64**

This model has the following Iris instances:

Name	Instance type
VECB2AMBAPVValue64	VECB2AMBAPVValue64

**Ports for VECB2AMBAPVValue64**

Port	Direction	Protocol	Description
amba_pv_ctrl_m	master	AMBAPVValue	AMBAPV portout.
amba_pv_data_m	master	AMBAPVValue64	AMBAPV portout.
vecb_s	slave	VECBProtocol	VECB port in.

**Parameters for VECB2AMBAPVValue64**

This component does not have any parameters.

3.426 VHT\_VIOBridge

Defined in `LISA/VHT_VIOBridge.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r1p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

## About VHT\_VIOBridge

This component is the Arm VHT virtual I/O interface. It relies on Python to work properly. The following list describes dependency resolution for supported operating systems:

### Windows:

If Python is installed at a custom location, make sure it is included in the PATH. Fast Models first searches the PATH and then falls back to use the Windows dynamic loader.

### Linux:

If Python is installed at a custom location, use the `LD_LIBRARY_PATH` environment variable. Fast Models first searches using this environment variable and then falls back to the default search path of the system dynamic linker.

## Ports for VHT\_VIOBridge

Port	Direction	Protocol	Description
pvbus_s	slave	PVBus	Target port for access to VIOBridge registers

## Parameters for VHT\_VIOBridge

### **vio\_basename**

Type: string

Default value: ""

### **vio\_path**

Type: string

Default value: ""

## 3.427 VHT\_VSIBridge

Defined in `LISA/VHT_VSIBridge.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r1p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

## About VHT\_VSIBridge

This component is the Arm VHT virtual stream interface. It relies on Python to work properly. The following list describes dependency resolution for supported operating systems:

**Windows:**

If Python is installed at a custom location, make sure it is included in the PATH. Fast Models first searches the PATH and then falls back to use the Windows dynamic loader.

**Linux:**

If Python is installed at a custom location, use the `LD_LIBRARY_PATH` environment variable. Fast Models first searches using this environment variable and then falls back to the default search path of the system dynamic linker.

**Ports for VHT\_VSIBridge**

Port	Direction	Protocol	Description
intr	master	Signal	Interrupt raising signal
pvbus_m	master	PVBus	Requester port for access to memory
pvbus_s	slave	PVBus	Completer port for access to VSIBridge registers

**Parameters for VHT\_VSIBridge****vsi\_basename**

Type: `string`

Default value: ""

**vsi\_idx**

Type: `uint32_t`

Default value: 0

**vsi\_path**

Type: `string`

Default value: ""

## 3.428 VHT\_VSocket

Defined in `LISA/VHT_VSocket.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
r1p0	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

**About VHT\_VSocket**

Arm VHT Virtual socket bridge interface.

### Ports for VHT\_VSocket

Port	Direction	Protocol	Description
pvbus_m	master	PVBus	Port for VSocket bridge to access external memory
pvbus_s	slave	PVBus	Target port for access to VSocket bridge registers

### Parameters for VHT\_VSocket

This component does not have any parameters.

## 3.429 Value64Logger

Defined in `LISA/Value64Logger.lisa`.

### About Value64Logger

Traces value activity.

### Iris and MTI instances for Value64Logger

This model has the following Iris instances:

Name	Instance type
Value64Logger	Value64Logger

This model has the following MTI trace components:

Name	Component type
Value64Logger	Value64Logger

### Ports for Value64Logger

Port	Direction	Protocol	Description
in	slave	Value_64	Input signal port.
out	master	Value_64	Output signal port.

### Parameters for Value64Logger

This component does not have any parameters.

## 3.430 ValueLogger

Defined in `LISA/ValueLogger.lisa`.

### About ValueLogger

Traces value activity.

### Iris and MTI instances for ValueLogger

This model has the following Iris instances:

Name	Instance type
ValueLogger	ValueLogger

This model has the following MTI trace components:

Name	Component type
ValueLogger	ValueLogger

### Ports for ValueLogger

Port	Direction	Protocol	Description
in	slave	Value	Input signal port.
out	master	Value	Output signal port.

### Parameters for ValueLogger

This component does not have any parameters.

## 3.431 Value\_Multiplexer

Defined in `LISA/Multiplexer.lisa`.

### About Value\_Multiplexer

Signal Multiplexer.

### Iris and MTI instances for Value\_Multiplexer

This model has the following Iris instances:

Name	Instance type
Value_Multiplexer	Value_Multiplexer

### Ports for Value\_Multiplexer

Port	Direction	Protocol	Description
input	slave	Value	-
output	master	Value	-
selector	slave	Value	-

### Parameters for Value\_Multiplexer

### diagnostics

Diagnostics.

Type: `uint8_t`

Default value: 2

## 3.432 VirtioBlockDevice

Defined in `LISA/VirtioBlockDevice.lisa`.

### About VirtioBlockDevice

VirtioBlockDevice implements a block device that can be accessed from the simulated OS if it has an appropriate driver. Similarly to the VirtioP9Device, this component is targeted primarily at Linux, which has a built-in virtio block driver. VirtioBlockDevice allows you to use a file on the host that you specify using the `image_path` parameter, as a hard drive in the simulated OS.

VirtioBlockDevice supports the legacy OASIS virtio specification.

Unlike the VirtioP9Device, you should not need to carry out any special setup to use VirtioBlockDevice on VE or Base platforms, because it is usually already included in the device trees. Set the `image_path` parameter to point to your image, and then on Linux it is available as a block device, usually `/dev/vda`, which you then use like any other hard drive.

### Iris and MTI instances for VirtioBlockDevice

This model has the following Iris instances:

Name	Instance type
VirtioBlockDevice	VirtioBlockDevice
VirtioBlockDevice.register_slave	PVBusSlave
VirtioBlockDevice.virtio_master	PVBusMaster

This model has the following MTI trace components:

Name	Component type
VirtioBlockDevice	VirtioBlockDevice
VirtioBlockDevice.register_slave	PVBusSlave
VirtioBlockDevice.virtio_master	PVBusMaster

### Ports for VirtioBlockDevice

Port	Direction	Protocol	Description
intr	master	Signal	Virtio device sets interrupt to signal completion.
pvbus	slave	PVBus	Virtio MMIO control/config/status registers.
virtio_m	master	PVBus	Virtio device performs DMA accesses via master.

## Parameters for VirtioBlockDevice

### **image\_path**

image file path.

Type: `string`

Default value: `N/A`

### **quiet**

Don't print warnings on malformed commands/descriptors.

Type: `bool`

Default value: `false`

### **read\_only**

Only allow device to be read.

Type: `bool`

Default value: `false`

### **secure\_accesses**

Make device generate transactions with NS=0.

Type: `bool`

Default value: `false`

### **transaction\_attributes**

Transaction attributes used by device.x0 - inner-shared real access.x1 - outer-shared real access.x2 - outer-shared debug access.

Type: `uint32_t`

Default value: 0

## 3.433 VirtioBlockDeviceMMIO

Defined in `LISA/VirtioBlockDeviceMMIO.lisa`.

### Changes in 11.31.15

The following parameters were added:

- `use_iommu_platform`

## About VirtioBlockDeviceMMIO

VirtioBlockDeviceMMIO supports both the legacy and v1.0 OASIS virtio specifications.



VirtioBlockDeviceMMIO is an evolution of [VirtioBlockDevice](#), which is also MMIO-based and has the same ports, but only supports the legacy OASIS virtio specification.

## Iris and MTI instances for VirtioBlockDeviceMMIO

This model has the following Iris instances:

Name	Instance type
VirtioBlockDeviceMMIO	<a href="#">VirtioBlockMMIO</a>
VirtioBlockDeviceMMIO.dma_master	<a href="#">PVBusMaster</a>

This model has the following MTI trace components:

Name	Component type
VirtioBlockDeviceMMIO	<a href="#">VirtioBlockMMIO</a>
VirtioBlockDeviceMMIO.dma_master	<a href="#">PVBusMaster</a>

## Ports for VirtioBlockDeviceMMIO

Port	Direction	Protocol	Description
intr	master	<a href="#">Signal</a>	Virtio device sets interrupt to signal completion.
pvbus	slave	<a href="#">PVBus</a>	Virtio MMIO control/config/status registers.
virtio_m	master	<a href="#">PVBus</a>	Virtio device performs DMA accesses via master.

## Parameters for VirtioBlockDeviceMMIO

### **enabled**

Enable or disable device. If disabled, device can be accessed, but will not be activated.

Type: `bool`

Default value: `false`

### **image\_path**

Image file path.

Type: `string`

Default value: `N/A`

### **quiet**

Don't print info or warnings (e.g. on malformed commands/descriptors).



Type: `bool`

Default value: `false`

### **`read_only`**

Only allow device to be read. If that parameter is set to false and the image file cannot be opened in RW mode, the model will try to work around it by opening the file in RO mode.

Type: `bool`

Default value: `false`

### **`secure_accesses`**

Make device generate transactions with NS=0.

Type: `bool`

Default value: `false`

### **`transaction_attributes`**

Transaction attributes used by device.x0 - inner-shared real access.x1 - outer-shared real access.x2 - outer-shared debug access.

Type: `uint32_t`

Default value: 0

### **`transport`**

Choose legacy or modern virtio transport, if not specified, modern transport is used.

Type: `string`

Default value: `"modern"`

### **`use_iommu_platform`**

Advertise `DEV_F_IOMMU_PLATFORM` (`VIRTIO_F_ACCESS_PLATFORM`); requires DMA via platform IOMMU.

Type: `bool`

Default value: `false`

## 3.434 VirtioNetMMIO

Defined in `LISA/VirtioNetMMIO.lisa`.

### Changes in 11.31.15

The following parameters were added:

- `num_queue_pairs`
- `use_ctrl_guest_offloads`
- `use_ctrl_mac_addr`
- `use_ctrl_rx`
- `use_ctrl_vq`
- `use_iommu_platform`
- `use_mq_rx_steering`
- `use_mtu_feature`

### About VirtioNetMMIO

This is a model of a virtual Ethernet virtio device over MMIO transport, supporting both the legacy and v1.0 OASIS virtio specifications. It provides much better network performance than the `SMSC_91C111` component, because it features host-assisted network acceleration. This means that it can offload packet processing operations from the simulated OS on the target, to the host side. These operations include:

- Checksum computation
- TX packet segmentation
- RX packet combination

If the target simulated Linux or Linux-derived OS has an appropriate virtio net driver, Arm recommends you use `VirtioNetMMIO` instead of `SMSC_91C111`.

Unlike `SMSC_91C111`, which must work with an external `HostBridge` component, `VirtioNetMMIO` has a built-in `HostBridge` sub-component. The parameters to control the `HostBridge` are described in the `VirtioNetMMIO` parameters table, with the `hostbridge` parameter sub-namespace.

To enable tracing of user-mode networking, which can help to debug networking issues, set the `FASTSIM_USERNET_DUMP` environment variable to any or all of the following values:

```
arpin,arpout,udpin,udpout,etherin,etherout,ipv4in,ipv4out,  
ipv4fragin,ipv4fragout,tcpin,tcpout,dhcpv4in,dhcpv4out
```

Take the following steps to set up this component in a virtual platform:

- Use a version of Linux that contains a virtio network driver.

- Add the following option to the Linux kernel configuration:

```
CONFIG_VIRTIO_NET=y
```

- Update the device tree to include the VirtioNetMMIO component, or specify it on the kernel command line, for example

```
virtio_mmio.device=0x10000@0x1c150000:76
```

The address range for both VE and Base platforms is 0x1c150000-0x1c15FFFF. The interrupt number is 44, or IRQ 76, for both VE and Base platforms.

- Select the hostbridge that you want to use to communicate with the host in the model:

```
virtio_net.hostbridge.userNetworking=true/false (User mode or TAP/TUN networking)
```

- Configure the networking environment, as described in [Configuring the networking environment for Linux](#).

### Example entries for DTS files

- Add this entry next to the corresponding virtio\_block or virtio\_p9 entry:

```
virtio_net@0150000 {  
    compatible = "virtio,mmio";  
    reg = <0x150000 0x1000>;  
    interrupts = <0x2c>;  
};
```

- Add this entry to the interrupt map:

```
<0 0 44 &gic 0 44 4>;
```

### See also

- [Configuring the networking environment for Linux](#)

### Iris and MTI instances for VirtioNetMMIO

This model has the following Iris instances:

Name	Instance type
VirtioNetMMIO	VirtioNetMMIO
VirtioNetMMIO.dma_master	PVBusMaster
VirtioNetMMIO.hostbridge	HostBridge

This model has the following MTI trace components:

Name	Component type
VirtioNetMMIO	VirtioNetMMIO
VirtioNetMMIO.dma_master	PVBusMaster

## Ports for VirtioNetMMIO

Port	Direction	Protocol	Description
intr	master	Signal	Virtio device sets interrupt to signal completion.
pvbust	slave	PVBus	Virtio MMIO control/config/status registers.
virtio_m	master	PVBus	Virtio device performs DMA accesses via master.

## Parameters for VirtioNetMMIO

### **checksum**

For checksum-offloaded packets. 'tx': checksum outgoings. 'rx': checksum incomings; 'all': both.

Type: `string`

Default value: `""`

### **enabled**

Enable or disable device. If disabled, device can be accessed, but will not be activated.

Type: `bool`

Default value: `false`

### **hostbridge.interfaceName**

Host Interface.

Type: `string`

Default value: `""`

### **hostbridge.userNetOptions**

Control options for UserNet TCP/IP, `tcp.xxx=xxx`, `ip.xxx=xxx`.

Type: `string`

Default value: `""`

### **hostbridge.userNetPorts**

Listening ports to expose in user-mode networking.

Type: `string`

Default value: `""`

### **hostbridge.userNetSubnet**

Virtual subnet for user-mode networking.

Type: `string`

Default value: `"172.20.51.0/24"`

### **hostbridge.userNetworking**

Enable user-mode networking.

Type: `bool`

Default value: `false`

### **mac\_address**

Device MAC address, if not specified, a random MAC address is generated.

Type: `string`

Default value: `""`

### **num\_queue\_pairs**

Number of RX/TX queue pairs to expose. 1 = single-queue (NET\_F\_MQ off).

Type: `int`

Default value: `1`

### **offload**

Offload TCP/UDP segmentation/receiving operations to host.

Type: `string`

Default value: `"gso, gro"`

### **secure\_accesses**

Make device generate transactions with NS=0.

Type: `bool`

Default value: `false`

### **transport**

Choose legacy or modern virtio transport, if not specified, modern transport is used.

Type: `string`

Default value: `"modern"`

**use\_ctrl\_guest\_offloads**

Enable guest offload control via control virtqueue and advertise VIRTIO\_NET\_F\_CTRL\_GUEST\_OFFLOADS – not used by default in reference platforms.

Type: `bool`

Default value: `false`

**use\_ctrl\_mac\_addr**

Allow MAC address updates via control virtqueue and advertise VIRTIO\_NET\_F\_CTRL\_MAC\_ADDR.

Type: `bool`

Default value: `false`

**use\_ctrl\_rx**

(Advanced) Enable RX mode control via control virtqueue and advertise VIRTIO\_NET\_F\_CTRL\_RX. Not used by default in reference platforms.

Type: `bool`

Default value: `false`

**use\_ctrl\_vq**

Enable virtio-net control virtqueue and advertise VIRTIO\_NET\_F\_CTRL\_VQ.

Type: `bool`

Default value: `false`

**use\_iommu\_platform**

Advertise DEV\_F\_IOMMU\_PLATFORM (VIRTIO\_F\_ACCESS\_PLATFORM); requires DMA via platform IOMMU.

Type: `bool`

Default value: `false`

**use\_mq\_rx\_steering**

(Advanced) Enable simple RX steering across multiqueue RX queues (round-robin). Not used by default in reference platforms.

Type: `bool`

Default value: `false`

**use\_mtu\_feature**

Advertise VIRTIO\_NET\_F\_MTU and populate mtu in config space.

Type: `bool`

Default value: `false`

## 3.435 VirtioP9Device

Defined in `LISA/VirtioP9Device.lisa`.

### About VirtioP9Device

This component implements a subset of the Plan 9 file protocol over a virtio transport. It enables you to access a directory on the host's filesystem within Linux, or another operating system that implements the protocol, running on a platform model.

It supports the legacy OASIS virtio specification.

### Limitations

VirtioP9Device implements a subset of the Linux 9P2000.L protocol. It has the following limitations:

- You can mount only one host directory per instance of the component.
- It supports a subset of 9P2000.L message types:
  - `Tversion`
  - `Tlopen`
  - `Tlcreate`
  - `Tgetattr`
  - `Tsetattr`
  - `Treaddir`
  - `Tmkdir`
  - `Tattach`
  - `Twalk`
  - `Tread`
  - `Twrite`
  - `Tclunk`
  - `Tremove`
  - `Trename`
  - On Linux hosts, it also supports `Treadlink` and `Tsymlink`.
- On Windows hosts:

- It ignores Unix permissions when writing files.
- It performs a simple mapping from Windows to Unix permissions when reading.
- Symbolic links appear as regular files.
- It does not perform writing, deleting, or renaming operations on a file that another process has open.

Setting up the VirtioP9Device component

Take the following steps to set up this component:

- Use a version of Linux that supports v9fs over virtio and virtio-mmio devices.
- Update the device tree to include the VirtioP9Device component, or specify it on the kernel command-line, as shown below. The address range for both VE and Base platforms is 0x1C140000-0x1C14FFFF. The interrupt number is 43, or IRQ 75, for both VE and Base platforms.
- Set the following parameter to the directory on the host that you want to mount in the model:

VE

motherboard.virtiop9device.root\_path

Base

bp.virtiop9device.root\_path

- On Linux, mount the host directory by using the following command in the model:

```
$ mount -t 9p -o trans=virtio,version=9p2000.L FM <mount point>
```

Example kernel command-line argument:

```
virtio_mmio.device=0x100000@0x1c140000:75
```

Example entry for DTS files, to add next to the corresponding virtio\_block entry:

```
virtio_p9@0140000 {
    compatible = "virtio,mmio";
    reg = <0x0 0x1c140000 0x0 0x1000>;
    interrupts = <0x0 0x2b 0x4>;
};
```

Iris and MTI instances for VirtioP9Device

This model has the following Iris instances:

Name	Instance type
VirtioP9Device	VirtioP9Device
VirtioP9Device.mmio_slave	PVBusSlave
VirtioP9Device.virtio_master	PVBusMaster

This model has the following MTI trace components:



Name	Component type
VirtioP9Device.mmio_slave	PVBusSlave
VirtioP9Device.virtio_master	PVBusMaster

### Ports for VirtioP9Device

Port	Direction	Protocol	Description
intr	master	Signal	Virtio device sets interrupt to signal completion.
pvbus	slave	PVBus	Virtio MMIO control/config/status registers.
virtio_m	master	PVBus	Virtio device performs DMA accesses via master.

### Parameters for VirtioP9Device

#### **mount\_tag**

mount tag.

Type: `string`

Default value: `"FM"`

#### **quiet**

Don't print warnings on malformed commands/descriptors.

Type: `bool`

Default value: `false`

#### **root\_path**

root directory path.

Type: `string`

Default value: `N/A`

#### **secure\_accesses**

Make device generate transactions with NS=0.

Type: `bool`

Default value: `false`

## 3.436 VirtioPCIBlockDevice

Defined in `LISA/VirtioPCIBlockDevice.lisa`.

### About VirtioPCIBlockDevice

VirtioPCIBlockDevice is similar to VirtioBlockDevice, except it is PCI-based instead of MMIO-based. It supports the legacy OASIS virtio specification.

This device requires:

- The following BARs in rising order:
  - A Bar of 4K for Config accesses
  - A Bar of 4K for the MSI-X table
  - A Bar of 4K for the MSI-X PBA
- Conventional interrupts to be supported
- An `msix_table_size` of 2

### Iris and MTI instances for VirtioPCIBlockDevice

This model has the following Iris instances:

Name	Instance type
VirtioPCIBlockDevice	VirtioPCIBlockDevice
VirtioPCIBlockDevice.register_slave	PVBusSlave
VirtioPCIBlockDevice.virtio_master	PVBusMaster

This model has the following MTI trace components:

Name	Component type
VirtioPCIBlockDevice	VirtioPCIBlockDevice
VirtioPCIBlockDevice.register_slave	PVBusSlave
VirtioPCIBlockDevice.virtio_master	PVBusMaster

### Ports for VirtioPCIBlockDevice

Port	Direction	Protocol	Description
client_s	slave	PCIDevice2ClientProtocol	Interrupts for MSI-X table entries.
intr	master	Signal	Virtio device sets interrupt to signal completion.
pvbus	slave	PVBus	Virtio pci/control/config/status registers.
virtio_m	master	PVBus	Virtio device performs DMA accesses via master.

### Parameters for VirtioPCIBlockDevice

#### **image\_path**

image file path.

Type: `string`

Default value: `N/A`

### **quiet**

Don't print warnings on malformed commands/descriptors.

Type: `bool`

Default value: `false`

### **read\_only**

Only allow device to be read.

Type: `bool`

Default value: `false`

### **secure\_accesses**

Make device generate transactions with NS=0.

Type: `bool`

Default value: `false`

### **transaction\_attributes**

Transaction attributes used by device.x0 - inner-shared real access.x1 - outer-shared real access.x2 - outer-shared debug access.

Type: `uint32_t`

Default value: 0

## 3.437 VirtioRNG

Defined in `LISA/VirtioRNG.lisa`.

### About VirtioRNG

VirtioRNG models a virtio entropy device as defined in the [Virtio 1.0 Specification](#). A virtual platform might need to integrate a VirtioRNG component to generate random numbers when:

- Linux or Android needs to generate kernel entropy. Hardware might do this using a timer, but this is not possible in the model because timers are not updated quickly enough.
- Security features are required, such as ssh.

## Integrate VirtioRNG into a platform

Integrate the VirtioRNG component by instantiating it in your board's LISA file and connecting it to the SoC virtio master bus and interrupt signal as follows:

```
// Instantiate components
composition {
    ...
    virtio_rng : VirtioRNG();
    ...
}

connection {
    ...
    // Find a suitable address space and connect it to the SoC's virtio_m bus
    busdecoder.pvbus_m_range[0x001C190000..0x001C19ffff] => virtio_rng.pvbus;
    virtio_rng.virtio_m => self.virtio_m;

    // Connect the IRQ to the GIC IRQ
    virtio_rng.intr => gic400.irqs[101];
    ...
}
```

To configure Linux or Android for VirtioRNG, use the following build parameters:

### Linux:

- CONFIG\_VIRTIO\_MMIO=y
- CONFIG\_HW\_RANDOM=y
- CONFIG\_HW\_RANDOM\_VIRTIO=y

### Android:

- --enable CONFIG\_VIRTIO\_MMIO
- --enable CONFIG\_HW\_RANDOM
- --enable CONFIG\_HW\_RANDOM\_VIRTIO

Use the following device tree parameters:

```
virtio_rng@1c190000 {
    compatible = "virtio,mmio";
    reg = <0x0 0x1c190000 0x0 0x200>;
    interrupts = <GIC_SPI 101 IRQ_TYPE_LEVEL_HIGH>;
};
```

Configure VirtioRNG using model parameters, for example:

```
-C "board.virtio_rng.enabled=1" \
-C "board.virtio_rng.seed=0" \
-C "board.virtio_rng.generator=2" \
-C "board.virtio_rng.diagnostics=4" \ # Optional
```

Use the following guest command line to test the integration:

```
// Generate random numbers
```

```
console/> cat /dev/hwrng
```

## Iris and MTI instances for VirtioRNG

This model has the following Iris instances:

Name	Instance type
VirtioRNG	<a href="#">VirtioEntropyMMIO</a>
VirtioRNG.dma_master	<a href="#">PVBUSMaster</a>

This model has the following MTI trace components:

Name	Component type
VirtioRNG	<a href="#">VirtioEntropyMMIO</a>
VirtioRNG.dma_master	<a href="#">PVBUSMaster</a>

## Ports for VirtioRNG

Port	Direction	Protocol	Description
intr	master	<a href="#">Signal</a>	Virtio device sets interrupt to signal completion.
pvbust	slave	<a href="#">PVBUS</a>	Virtio control/config/status registers.
virtio_m	master	<a href="#">PVBUS</a>	Virtio device performs DMA accesses via master.

## Parameters for VirtioRNG

### diagnostics

Prints debug information: 0 - disabled; 1 - generated seed and device; 4 - generated seed, device and generated numbers.

Type: `uint32_t`

Default value: 0

### enabled

Enable or disable device. If disabled, device can be accessed, but will not be activated.

Type: `bool`

Default value: `false`

### generator

User-defined generator: 0 = xorshiftstar; 1 = rand48; 2 = mersenne;.

Type: `uint32_t`

Default value: 0

**secure\_accesses**

Make device generate transactions with NS=0 [not supported].

Type: `bool`

Default value: `false`

**seed**

User-defined seed: 0 = uses a random seed; > 0 = user-defined fixed seed value.

Type: `uint32_t`

Default value: 0

**transport**

Choose legacy [not supported] or modern virtio transport. If not specified, modern transport is used.

Type: `string`

Default value: `"modern"`

3.438 VirtualEthernet2SystemC

Defined in `examples/SystemCExport/Bridges/VirtualEthernet2SystemC.lisa`.

About VirtualEthernet2SystemC

VirtualEthernet to SystemC Converter.

Iris and MTI instances for VirtualEthernet2SystemC

This model has the following Iris instances:

Name	Instance type
VirtualEthernet2SystemC	VirtualEthernet2SystemC

Ports for VirtualEthernet2SystemC

Port	Direction	Protocol	Description
virtualethernet_m	master	SC_VirtualEthernet	-
virtualethernet_s	slave	VirtualEthernet	-

Parameters for VirtualEthernet2SystemC

This component does not have any parameters.

## 3.439 VirtualEthernetCrossover

Defined in `LISA/VirtualEthernetCrossover.lisa`.

### About VirtualEthernetCrossover

This component implements two VirtualEthernet slave ports and enables you to connect two VirtualEthernet master ports. It forwards data received on one port to the other port without delay.

### Iris and MTI instances for VirtualEthernetCrossover

This model has the following Iris instances:

Name	Instance type
VirtualEthernetCrossover	VirtualEthernetCrossover

This model has the following MTI trace components:

Name	Component type
VirtualEthernetCrossover	VirtualEthernetCrossover

### Ports for VirtualEthernetCrossover

Port	Direction	Protocol	Description
deva	slave	VirtualEthernet	Slave port for connecting to a VirtualEthernet master.
devb	slave	VirtualEthernet	Slave port for connecting to a VirtualEthernet master.

### Parameters for VirtualEthernetCrossover

This component does not have any parameters.

## 3.440 VirtualEthernetHub3

Defined in `LISA/VirtualEthernetHub3.lisa`.

### About VirtualEthernetHub3

3 Port Ethernet Hub.

### Iris and MTI instances for VirtualEthernetHub3

This model has the following Iris instances:

Name	Instance type
VirtualEthernetHub3	VirtualEthernetHub3

## Ports for VirtualEthernetHub3

Port	Direction	Protocol	Description
deva	slave	VirtualEthernet	Slave port for connecting to a VirtualEthernet master.
devb	slave	VirtualEthernet	Slave port for connecting to a VirtualEthernet master.
devc	slave	VirtualEthernet	Slave port for connecting to a VirtualEthernet master.

## Parameters for VirtualEthernetHub3

This component does not have any parameters.

## 3.441 VisEventRecorder

Defined in `LISA/VisEventRecorder.lisa`.

### Recording

The following command creates an ASCII file `rec.txt` and enables recording. This file can directly be used for playback.

```
./isim_system -a image.axf -C visualisation.recorder.recordingFileName=rec.txt
```

You can select the time base for the time stamps of the recorded events. The default is a 100MHz counter (10ns simulated time resolution) which usually works for all systems. To be able to correlate timestamps to the instruction counter, set the time base to the clock frequency of the CPU, but this is not necessary for an exact recording or playback. The time base should be higher than CPU frequency / 100 (typical quantum size). To set the recording time base set the `recordingTimeBase` parameter.

### Playback

The following command enables the playback of all GUI input events previously recorded in file `rec.txt`. The time base of the timestamps is always taken from the file (see T record). Interactive user input is still possible and interactive events and recorded events are mixed.

```
./isim_system -a image.axf -C visualisation.recorder.playbackFileName=rec.txt
```



#### Note

It is possible to enable recording and playback at the same time. This makes it possible to check whether a playback session is reproducible or to extend a previously recorded session by appending events. To do this, remove the QUIT event at the end. This is also useful to check the timing accuracy of the playback/recording timestamps.

```
./isim system -a image.axf -C  
visualisation.recorder.playbackFileName=rec.txt -C  
visualisation.recorder.playbackFileName=rec.txt
```



To enable verbose messages, use the `verbose` parameter with the following values:

1

Print all events while they are recorded/played back.

2

Print also maintenance information of the internal `ClockTimers`. The default is disabled (0).

To disable instruction count checking (message 'instruction count differs'), set parameter `checkInstructionCount` to 0. The default is enabled.

## Integration

This component is intended to be a subcomponent of a visualisation component, for example a component that instantiates a `Visualisation` object using `createVisualisation()`. The integration is pretty light weight:

- Wire up the `control` and `ticks` ports.
- Use `control.registerVisRegion(regionPointer, regionName)` to register all relevant `VisRegion` pointers. You only need to register the regions that are used in the `processMessages()` function to identify a region by pointer.
- Call `control.putEvent()` for all `visEvents` as they come in regardless of where they come from, usually from `processMessages()`.
- Call `control.getEvent()` to retrieve recorded events (always called directly or indirectly by the callback (master) behavior `control.processEvents()`).

## Iris and MTI instances for VisEventRecorder

This model has the following Iris instances:

Name	Instance type
<code>VisEventRecorder</code>	<code>VisEventRecorder</code>
<code>VisEventRecorder.playbackDivider</code>	<code>ClockDivider</code>
<code>VisEventRecorder.playbackTimer</code>	<code>ClockTimerThread</code>
<code>VisEventRecorder.playbackTimer.timer</code>	<code>ClockTimerThread64</code>
<code>VisEventRecorder.playbackTimer.timer.thread</code>	<code>SchedulerThread</code>
<code>VisEventRecorder.playbackTimer.timer.thread_event</code>	<code>SchedulerThreadEvent</code>
<code>VisEventRecorder.recordingDivider</code>	<code>ClockDivider</code>

This model has the following MTI trace components:

Name	Component type
<code>VisEventRecorder.playbackDivider</code>	<code>ClockDivider</code>
<code>VisEventRecorder.recordingDivider</code>	<code>ClockDivider</code>

## Ports for VisEventRecorder

Port	Direction	Protocol	Description
<code>control</code>	slave	<code>VisEventRecorderProtocol</code>	The visualisation component controls the recorder through this port.

Port	Direction	Protocol	Description
ticks	slave	InstructionCount	Allow VisEventRecorder to get tick count from a core.

## Parameters for VisEventRecorder

### **checkInstructionCount**

check instruction count in recording file against actual instruction count during playback.

Type: `bool`

Default value: `true`

### **playbackFileName**

playback filename (empty string disables playback).

Type: `string`

Default value: `""`

### **recordingFileName**

recording filename (empty string disables recording).

Type: `string`

Default value: `""`

### **recordingTimeBase**

timebase in 1/s (relative to the master clock (e.g. 100000000 means 10 nanoseconds resolution simulated time for a 1Hz master clock)) to be used for recording (higher values -> higher time resolution, playback time base is always taken from the playback file).

Type: `uint64_t`

Default value: `100000000`

### **verbose**

enable verbose messages (1=normal, 2=even more).

Type: `uint32_t`

Default value: `0`

## 3.442 Visualisation\_sdl2

Defined in `LISA/visualisation_sdl2.lisa`.

### About Visualisation\_sdl2

Display window for VE using sdl2 Visualisation library.

### Iris and MTI instances for Visualisation\_sdl2

This model has the following Iris instances:

Name	Instance type
Visualisation_sdl2	Visualisation_sdl2
Visualisation_sdl2.recorder	VisEventRecorder
Visualisation_sdl2.recorder.playbackDivider	ClockDivider
Visualisation_sdl2.recorder.playbackTimer	ClockTimerThread
Visualisation_sdl2.recorder.playbackTimer.timer	ClockTimerThread64
Visualisation_sdl2.recorder.playbackTimer.timer.thread	SchedulerThread
Visualisation_sdl2.recorder.playbackTimer.timer.thread_event	SchedulerThreadEvent
Visualisation_sdl2.recorder.recordingDivider	ClockDivider

This model has the following MTI trace components:

Name	Component type
Visualisation_sdl2.recorder.playbackDivider	ClockDivider
Visualisation_sdl2.recorder.recordingDivider	ClockDivider

### Ports for Visualisation\_sdl2

Port	Direction	Protocol	Description
c0_core_freq	slave	ValueState	-
c10_core_freq	slave	ValueState	-
c11_core_freq	slave	ValueState	-
c12_core_freq	slave	ValueState	-
c13_core_freq	slave	ValueState	-
c14_core_freq	slave	ValueState	-
c15_core_freq	slave	ValueState	-
c1_core_freq	slave	ValueState	-
c2_core_freq	slave	ValueState	-
c3_core_freq	slave	ValueState	-
c4_core_freq	slave	ValueState	-
c5_core_freq	slave	ValueState	-
c6_core_freq	slave	ValueState	-
c7_core_freq	slave	ValueState	-
c8_core_freq	slave	ValueState	-

Port	Direction	Protocol	Description
c9_core_freq	slave	ValueState	-
clock_50Hz	slave	ClockSignal	-
cluster0_ticks	slave	InstructionCount	-
cluster10_ticks	slave	InstructionCount	-
cluster11_ticks	slave	InstructionCount	-
cluster12_ticks	slave	InstructionCount	-
cluster13_ticks	slave	InstructionCount	-
cluster14_ticks	slave	InstructionCount	-
cluster15_ticks	slave	InstructionCount	-
cluster1_ticks	slave	InstructionCount	-
cluster2_ticks	slave	InstructionCount	-
cluster3_ticks	slave	InstructionCount	-
cluster4_ticks	slave	InstructionCount	-
cluster5_ticks	slave	InstructionCount	-
cluster6_ticks	slave	InstructionCount	-
cluster7_ticks	slave	InstructionCount	-
cluster8_ticks	slave	InstructionCount	-
cluster9_ticks	slave	InstructionCount	-
cluster_freq	slave	ValueState	-
keyboard	master	KeyboardStatus	-
lcd_layout	master	LCDLayoutInfo	-
lcd	slave	LCD	-
mcp_freq	slave	ValueState	-
mcp_ticks	slave	InstructionCount	-
mouse	master	MouseStatus	-
poreset	master	Signal	-
scp_freq	slave	ValueState	-
scp_ticks	slave	InstructionCount	-
sys_temperature	slave	ValueState	-
touch_screen	master	MouseStatus	-

## Parameters for Visualisation\_sdl2

### **cluster0\_name**

Cluster0 name.

Type: string

Default value: ""

### **cluster10\_name**

Cluster10 name.

Type: `string`

Default value: `""`

**cluster11\_name**

Cluster11 name.

Type: `string`

Default value: `""`

**cluster12\_name**

Cluster12 name.

Type: `string`

Default value: `""`

**cluster13\_name**

Cluster13 name.

Type: `string`

Default value: `""`

**cluster14\_name**

Cluster14 name.

Type: `string`

Default value: `""`

**cluster15\_name**

Cluster15 name.

Type: `string`

Default value: `""`

**cluster1\_name**

Cluster1 name.

Type: `string`

Default value: `""`

**cluster2\_name**

Cluster2 name.

Type: `string`

Default value: `""`

**cluster3\_name**

Cluster3 name.

Type: `string`

Default value: `""`

**cluster4\_name**

Cluster4 name.

Type: `string`

Default value: `""`

**cluster5\_name**

Cluster5 name.

Type: `string`

Default value: `""`

**cluster6\_name**

Cluster6 name.

Type: `string`

Default value: `""`

**cluster7\_name**

Cluster7 name.

Type: `string`

Default value: `""`

**cluster8\_name**

Cluster8 name.

Type: `string`

Default value: `""`

**cluster9\_name**

Cluster9 name.

Type: `string`

Default value: `""`

### **`css_spec`**

Platform specification displayed in window title.

Type: `string`

Default value: `"Columbus mid"`

### **`diagnostics`**

Diagnostics.

Type: `uint32_t`

Default value: `0`

### **`disable_visualisation`**

Enable/disable visualisation.

Type: `bool`

Default value: `false`

### **`display_object`**

Display objects: LCD only (1), status only (2), or both (0).

Type: `uint32_t`

Default value: `0`

### **`display_poreset_button`**

Display power-on reset button.

Type: `bool`

Default value: `false`

### **`idler.delay_ms`**

Determines the period, in milliseconds of real time, between `gui_callback()` calls.

Type: `uint32_t`

Default value: `50`

### **`idler.has_gui`**

Determines the period, in milliseconds of real time, between `gui_callback()` calls.

Type: `uint32_t`

Default value: 50

### **is\_heterogeneous\_cluster**

Is Heterogeneous cluster.

Type: `bool`

Default value: `false`

### **lcd\_height\_param**

LCD Height.

Type: `uint32_t`

Default value: 600

### **lcd\_width\_param**

LCD Width.

Type: `uint32_t`

Default value: 800

### **mcp\_name**

MCP name.

Type: `string`

Default value: "MCP: Cortex-M7"

### **num\_cps**

Number of Control Processors.

Type: `uint32_t`

Default value: 0x1

### **per\_core\_clock**

Per-core clock connection.

Type: `bool`

Default value: `false`

### **platform\_name**

Platform Name.



Type: `string`

Default value: `"Application Processors"`

### **`rate_limit-enable`**

Rate limit simulation.

Type: `bool`

Default value: `false`

### **`recorder.checkInstructionCount`**

check instruction count in recording file against actual instruction count during playback.

Type: `bool`

Default value: `true`

### **`recorder.playbackDivider.div`**

Clock Rate Divider. This parameter is not exposed via Iris and can only be set in LISA.

Type: `uint64_t`

Default value: `1`

### **`recorder.playbackDivider.mul`**

Clock Rate Multiplier. This parameter is not exposed via Iris and can only be set in LISA.

Type: `uint64_t`

Default value: `1`

### **`recorder.playbackFileName`**

playback filename (empty string disables playback).

Type: `string`

Default value: `""`

### **`recorder.recordingDivider.div`**

Clock Rate Divider. This parameter is not exposed via Iris and can only be set in LISA.

Type: `uint64_t`

Default value: `1`

### **`recorder.recordingDivider.mul`**

Clock Rate Multiplier. This parameter is not exposed via Iris and can only be set in LISA.

Type: `uint64_t`

Default value: 1

### **recorder.recordingFileName**

recording filename (empty string disables recording).

Type: `string`

Default value: ""

### **recorder.recordingTimeBase**

timebase in 1/s (relative to the master clock (e.g. 100000000 means 10 nanoseconds resolution simulated time for a 1Hz master clock)) to be used for recording (higher values -> higher time resolution, playback time base is always taken from the playback file).

Type: `uint64_t`

Default value: 100000000

### **recorder.verbose**

enable verbose messages (1=normal, 2=even more).

Type: `uint32_t`

Default value: 0

### **scp\_name**

SCP name.

Type: `string`

Default value: "SCP: Cortex-M7"

### **shutdown\_pixel\_enable**

Shutdown pixel enable. Use to trigger simulation shutdown when a specific pixel reaches the given target RGB value.

Type: `bool`

Default value: `false`

### **shutdown\_pixel\_rgb**

Shutdown pixel target RGB value 0xRRGGBB.

Type: `uint32_t`

Default value: 0xFFFFFF

**shutdown\_pixel\_x**

Shutdown pixel X co-ordinate (0 is left).

Type: uint32\_t

Default value: 0

**shutdown\_pixel\_y**

Shutdown pixel Y co-ordinate (0 is top).

Type: uint32\_t

Default value: 0

**trap\_key**

Trap key that works with left Ctrl to toggle mouse display.

Type: uint32\_t

Default value: 74

**window\_title**

Window title(%cpu% will be replaced by css\_spec).

Type: string

Default value: "Fast Models - %cpu%"

3.443 WarningMemory

Defined in LISA/WarningMemory.lisa.

About WarningMemory

Memory that prints warnings, and **RAZ**/WIs or aborts.

Iris and MTI instances for WarningMemory

This model has the following Iris instances:

Name	Instance type
WarningMemory	WarningMemory

Ports for WarningMemory

Port	Direction	Protocol	Description
pdbus	slave	PVBus	Bus slave interface

## Parameters for WarningMemory

### **abort\_on\_reads**

Abort on reads (read 0 if false).

Type: `bool`

Default value: `false`

### **abort\_on\_writes**

Abort on writes (ignore if false).

Type: `bool`

Default value: `false`

### **read\_data**

Data to return on reads, if not aborting.

Type: `uint64_t`

Default value: `0`

### **warn\_on\_reads**

Warn on reads.

Type: `bool`

Default value: `true`

### **warn\_on\_writes**

Warn on writes.

Type: `bool`

Default value: `true`

### **warning**

Warning string.

Type: `string`

Default value: `"Invalid access"`

## 3.444 WideAndGate

Defined in `LISA/WideAndGate.lisa`.

### About WideAndGate

And Gate with up to 8 inputs.

### Iris and MTI instances for WideAndGate

This model has the following Iris instances:

Name	Instance type
WideAndGate	<a href="#">WideAndGate</a>

### Ports for WideAndGate

Port	Direction	Protocol	Description
input	slave	Signal	-
output	master	Signal	-

### Parameters for WideAndGate

#### diagnostics

Diagnostics.

Type: `uint32_t`

Default value: 0

## 3.445 WideOrGate

Defined in `LISA/WideOrGate.lisa`.

### About WideOrGate

Or Gate with up to 8 inputs.

### Iris and MTI instances for WideOrGate

This model has the following Iris instances:

Name	Instance type
WideOrGate	<a href="#">WideOrGate</a>

### Ports for WideOrGate

Port	Direction	Protocol	Description
input	slave	Signal	-

Port	Direction	Protocol	Description
output	master	Signal	-

## Parameters for WideOrGate

### diagnostics

Diagnostics.

Type: uint32\_t

Default value: 0

## 3.446 WideOrGate\_12x4

Defined in LISA/WideOrGate\_12x4.lisa.

### About WideOrGate\_12x4

Or Gate with up to 48 inputs and support to num\_cores.

### Iris and MTI instances for WideOrGate\_12x4

This model has the following Iris instances:

Name	Instance type
WideOrGate_12x4	WideOrGate_12x4

## Ports for WideOrGate\_12x4

Port	Direction	Protocol	Description
input	slave	Signal	-
output1	master	Signal	-
output2	master	Signal	-
output3	master	Signal	-

## Parameters for WideOrGate\_12x4

### diagnostics

Diagnostics.

Type: uint32\_t

Default value: 0

### max\_cores\_per\_cluster

Max number of cores per cluster.

Type: uint32\_t

Default value: 4

### **num\_cores**

Number of cores in cluster.

Type: uint32\_t

Default value: 4

## 3.447 v7VGICConfig2SystemC

Defined in `examples/SystemCExport/Bridges/v7VGICConfig2SystemC.lisa`.

### About v7VGICConfig2SystemC

Converts v7\_vgic\_configuration\_protocol to SystemC.

### Iris and MTI instances for v7VGICConfig2SystemC

This model has the following Iris instances:

Name	Instance type
v7VGICConfig2SystemC	v7VGICConfig2SystemC

### Ports for v7VGICConfig2SystemC

Port	Direction	Protocol	Description
all_received	master	AMBAPVSignal	Called when all other values have been set in opposite bridge.
cpu_interface_number_m	master	AMBAPVValue64	To SystemC.
inout_cluster_number_m	master	AMBAPVValue64	To SystemC.
inout_cpu_number_in_cluster_m	master	AMBAPVValue64	To SystemC.
manager_id_m	master	AMBAPVValue64	To SystemC.
manager_id_mask_m	master	AMBAPVValue64	To SystemC.
number_of_cores_m	master	AMBAPVValueState	To SystemC.
response_s	slave	AMBAPVSignal	From SystemC.
v7_vgic_config_s	slave	v7_VGIC_Configuration_Protocol	v7_VGIC_Configuration_Protocol input.

### Parameters for v7VGICConfig2SystemC

This component does not have any parameters.

## 3.448 v7\_VGIC

Defined in `LISA/v7_VGIC.lisa`.

This model supports the following revisions of the IP at the given quality levels:

Revision	Quality level
N/A	Full support

For an explanation of the quality levels, see [Quality level definitions](#).

### About v7\_VGIC

System VGIC architecture version v7.

### Iris and MTI instances for v7\_VGIC

This model has the following Iris instances:

Name	Instance type
v7_VGIC	<a href="#">v7_VGIC</a>
v7_VGIC.vgic_bus_slave	<a href="#">PVBusSlave</a>

This model has the following MTI trace components:

Name	Component type
v7_VGIC	<a href="#">v7_VGIC</a>
v7_VGIC.vgic_bus_slave	<a href="#">PVBusSlave</a>

### Ports for v7\_VGIC

Port	Direction	Protocol	Description
cfgsdisable	slave	<a href="#">Signal</a>	Disable write access to some GIC registers.
configuration	slave	<a href="#">v7_VGIC_Configuration_Protocol</a>	Configure the mapping of the core number (from ManagerID) to the core interface number.
fiq_in	slave	<a href="#">Signal</a>	FIQ inputs.
fiq_out	master	<a href="#">Signal</a>	FIQ outputs.
irq_in	slave	<a href="#">Signal</a>	IRQ inputs.
irq_out	master	<a href="#">Signal</a>	IRQ outputs.
ppi_core0	slave	<a href="#">Signal</a>	Private peripheral interrupts (ID16-ID31) for cpu 0.
ppi_core1	slave	<a href="#">Signal</a>	Private peripheral interrupts (ID16-ID31) for cpu 1.
ppi_core2	slave	<a href="#">Signal</a>	Private peripheral interrupts (ID16-ID31) for cpu 2.
ppi_core3	slave	<a href="#">Signal</a>	Private peripheral interrupts (ID16-ID31) for cpu 3.
ppi_core4	slave	<a href="#">Signal</a>	Private peripheral interrupts (ID16-ID31) for cpu 4.
ppi_core5	slave	<a href="#">Signal</a>	Private peripheral interrupts (ID16-ID31) for cpu 5.
ppi_core6	slave	<a href="#">Signal</a>	Private peripheral interrupts (ID16-ID31) for cpu 6.
ppi_core7	slave	<a href="#">Signal</a>	Private peripheral interrupts (ID16-ID31) for cpu 7.



Port	Direction	Protocol	Description
pvbus_s	slave	PVBus	Bus port for accessing distributor registers.
reporting_interface	slave	VGICReportingProtocol	Logging interface.
reset_signal	slave	Signal	Reset signal input.
spi	slave	Signal	SPI inputs.
vfiq_out	master	Signal	Virtual FIQ outputs.
virq_out	master	Signal	Virtual IRQ outputs.
wakeup_fiq	master	Signal	Wakeup signal for FIQ.
wakeup_irq	master	Signal	Wakeup signal for IRQ.

## Parameters for v7\_VGIC

### **core-impl-id**

Implementation ID to present for the cores.

Type: `uint32_t`

Default value: `0x3902043B`

### **dist-impl-id**

Implementation ID to present for the distributor.

Type: `uint32_t`

Default value: `0x3902043B`

### **enable\_log\_errors**

Enable logging of errors.

Type: `bool`

Default value: `false`

### **enable\_log\_fatal**

Enable logging of fatal errors.

Type: `bool`

Default value: `false`

### **enable\_log\_warnings**

Enable logging of warnings.

Type: `bool`

Default value: `false`

**enabled**

Enable the component. If it is disabled then all register writes will have no effect.

Type: `bool`

Default value: `true`

**number-of-cores**

number of core interfaces to present.

Type: `uint32_t`

Default value: 8

**number-of-ints**

number of interrupt pins.

Type: `uint32_t`

Default value: 224

**number-of-lrs**

number of list registers.

Type: `uint32_t`

Default value: 64

**vgic-version**

Version number of the VGIC interface.

Type: `uint32_t`

Default value: 11

## 3.449 v8EmbeddedCrossTrigger\_Matrix

Defined in `LISA/v8EmbeddedCrossTrigger.lisa`.

### About v8EmbeddedCrossTrigger\_Matrix

This is a model of a platform-level Cross Trigger Matrix (CTM) for connection to the Cross Trigger Interface (CTI) ports provided on Armv8-A processors in Fast Models. The combination of the CTI and the CTM provides an architectural model of the Coresight embedded triggering system.

A single instance of the `v8EmbeddedCrossTrigger_Matrix` component supports up to four clusters, each containing four cores. For example:

```
cluster0.cti[0] => v8ect.cti[0];
cluster0.cti[1] => v8ect.cti[1];
cluster0.cti[2] => v8ect.cti[2];
cluster0.cti[3] => v8ect.cti[3];
...
cluster3.cti[0] => v8ect.cti[12];
cluster3.cti[1] => v8ect.cti[13];
cluster3.cti[2] => v8ect.cti[14];
cluster3.cti[3] => v8ect.cti[15];
```

## Iris and MTI instances for `v8EmbeddedCrossTrigger_Matrix`

This model has the following Iris instances:

Name	Instance type
<code>v8EmbeddedCrossTrigger_Matrix</code>	<code>EmbeddedCT</code>

## Ports for `v8EmbeddedCrossTrigger_Matrix`

Port	Direction	Protocol	Description
<code>cti</code>	slave	<code>v8EmbeddedCrossTrigger_controlprotocol</code>	-

## Parameters for `v8EmbeddedCrossTrigger_Matrix`

### **`has_CTIAUTHSTATUS`**

Has the optional CTIAUTHSTATUS register.

Type: `bool`

Default value: `true`

### **`has_CTIDEVID_INOUT`**

Has the option of input gate in cross trigger matrix.

Type: `bool`

Default value: `true`

### **`number-of-channels`**

Number of channels in cross trigger matrix.

Type: `uint8_t`

Default value: 4

## 4. Plug-ins for Fast Models

This chapter describes the plug-ins that are available for Fast Models.

Prebuilt plug-ins can be found at `$PVLIB_HOME/plugins/<OS_compiler>/`. The source code for some of these plug-ins is provided as programming examples, under `$PVLIB_HOME/examples/MTI/`.

### 4.1 Loading a plug-in

The method of loading a plug-in depends on the type of model being used.

#### 4.1.1 --plugin command-line option

To load a plug-in, use the `--plugin <path_to_plugin>/<plugin_name>` option.

To load more than one plug-in, use multiple `--plugin` options.

Specify plug-in parameters using one or more `-c` options when launching the model, or use a configuration file.

For example the following command:

- Loads the TarmacTrace plug-in.
- Sets the TarmacTrace `trace-file` parameter.

```
./isim_system --plugin $plugin_dir/TarmacTrace.so -C TRACE.TarmacTrace.trace-  
file=tTrace.log
```

#### 4.1.2 scx::scx\_load\_plugin() method

Fast Models that are exported to SystemC can call the method `scx::scx_load_plugin()` to hard code the path to the plug-in, before calling `sc_start()`.

For example:

```
scx::scx_load_plugin("$PVLIB_HOME/plugins/Linux64_GCC-10.3/TarmacTrace.so");
```

For more information, see [scx API](#).

### 4.1.3 FM\_PLUGINS environment variable

As an alternative to loading a plug-in using `--plugin`, you can use the environment variable `FM_PLUGINS`.

Plug-ins listed in `FM_PLUGINS` are implicitly loaded. `FM_PLUGINS` must be set to the full path of the plug-in. For example:

```
FM_PLUGINS=$FASTMODELS_HOME/plugins/Linux64_GCC-12.3/TarmacTrace.so
```

To load multiple plug-ins at the same time, separate them with semicolons. For example:

```
FM_PLUGINS=$FASTMODELS_HOME/plugins/Linux64_GCC-12.3/TarmacTrace.so;$FASTMODELS_HOME/
plugins/Linux64_GCC-12.3/GenericTrace.so
```

You can also load the same plug-in multiple times. Give each instance a name by adding a prefix `<instancename>=` to each plug-in path or paths.

To set parameters for a plug-in loaded using `FM_PLUGINS`, use this syntax:

```
|<param1=A>||<param2=B>||<paramN=C>|<path_to_plugin.so>
```

Parameters must be specified without the prefix that would be used on the command line, for example, `end-instruction-count` instead of `TRACE.TarmacTrace.end-instruction-count`.

For example:

```
FM_PLUGINS=|trace-file=/home/work/trace.txt||end-instruction-count=1000|
$FASTMODELS_HOME/plugins/Linux64_GCC-12.3/TarmacTrace.so
```

To load multiple plug-ins, each with their own parameters, use this syntax:

```
|<param1=A>||<param2=B>|<plugin1>=<path_to_plugin1.so>;|<param1=C>||<param2=D>|
<plugin2>=<path_to_plugin2.so>
```

Parameters apply to the plug-in they are grouped with. For example, `param1=A` and `param2=B` apply to `plugin1`. `param1=C` and `param2=D` apply to `plugin2`.



Note

- Do not specify a pipe character `|` at the end of the environment variable value.
- Separate individual parameters with two pipe characters `||`, not one.
- Separate the plug-in from the parameters using a single pipe character `|`.
- Do not specify quotes around paths that contain spaces, or escape spaces. Spaces are resolved automatically.
- Do not specify environment variables within this environment variable. When setting `FM_PLUGINS` in the shell, you can use environment variables because the shell replaces them with their expanded value.
- Specify the parameters before you specify the plug-in.

- On Windows, you can either set the environment variable using the **Advanced System Settings** window, or if you are using the command line, you might need to escape each pipe character, for example `^|`.
  - The parameter values that you set using this environment variable are not displayed by the `--list-params` or `-l` command-line options. This is because the environment variable is processed at a later stage, when launching the simulation.
- 

## 4.2 Customizing a plug-in

You can customize the behavior of a plug-in using parameters.

Plug-in parameters are set in the same way as model parameters. They have the following format:

```
-C <PLUGIN_TYPE>.<plugin_name>.<parameter>=<value>
```

The `PLUGIN_TYPE` value varies depending on the plug-in type, for example:

```
-C TRACE.TarmacTrace.trace-file=trace.log
```

```
-C CRYPTO.Crypto.verbose=1
```

## 4.3 ArchMsgTrace

The Architecture Message Trace plug-in prints warning and error messages to `stdout` or to a file when software performs operations that are not recommended, for instance because they are unpredictable.

The plug-in connects to all trace sources that have the `ArchMsg` prefix, which are normal MTI trace sources, but with a specific format. These trace sources can also be used with the `GenericTrace` plug-in, but the `ArchMsgTrace` plug-in has extra capabilities.

When the model emits an `ArchMsg` trace event, `ArchMsgTrace` outputs a message in the format:

```
category: component.hierarchy.name: ...
```

The trace sources have names of the form:

```
component.hierarchy.ArchMsg.category.name[#supplementalEventName]
```

where:

- `category` is usually `Warning`, `Error`, or `Info`.

- `name` is a short string that uniquely identifies the condition.
- `supplementalEventName` is an optional identifier for a supplemental event, which is an event that provides more information about the initial event. For example, if a cache contains mismatching attributes, which triggers an `ArchMsg` trace event, a supplemental event might be emitted for each cache line affected.

The trace source can also include a line that defines a more human-readable description of the event. This line can contain fields which `ArchMsgTrace` replaces in the output string with values from the trace source.

`ArchMsgTrace` can be configured to suppress:

- Specific trace sources.
- Specific categories.
- Repeated events of the same type.

To suppress specific trace sources or categories, use a whitespace-separated list of patterns, optionally including wildcards (`*` and `?`).

Repeated events can only be suppressed if the `ArchMsg` trace source declares a key field. `ArchMsgTrace` searches for the key field in the following way:

- It looks for the string `"\nPRIMARY KEY <key-field-name>"` in the description of the trace source and uses that field name if the string exists.
- If not found, it looks for a field named `"KEY"`.
- Otherwise, the `ArchMsg` trace source has no key field and cannot be suppressed.

If the `suppress_repeated` parameter is true, the plug-in suppresses repeated events for the same trace source that have the same key field value. For example, the key field might represent the PC and so repeated events for the same PC can be suppressed.



To see a list of all possible `ArchMsg` trace sources that the model can emit, run it with the `ListTraceSources` plug-in. Then search the output for trace sources with the `ArchMsg` prefix.

---

Some examples of `ArchMsg` trace sources are:

**`ArchMsg.Error.BusActiveDuringReset`**

A transaction was received at the bus slave port whilst reset was asserted.

**`ArchMsg.Warning.cache_contents_unknown`**

Execution that depends on unknown cache contents.

**`ArchMsg.Warning.warning_atomic_to_unsupported_memory`**

Atomic access to an unsupported memory type.

**`ArchMsg.Warning.decode_unpred_other`**

Use of unpredictable instruction.

**ArchMsg.Warning.recursive\_exception**

Recursive exception.

**Related information**

[ListTraceSources](#) on page 6370

**4.3.1 ArchMsgTrace - parameters**

Each parameter is prefixed with `TRACE.ArchMsgTrace`, for example:

```
TRACE.ArchMsgTrace.exit_on_first_output
```

Name	Type	Default value	Allowed values	When set	Description
exit_on_first_output	Bool	0x0	true, false	Init time	Exit the simulation process after the first message has been written
filter_tags	String	"ALL"	""	Init time	Space-separated list of tags that are matched against the tag(s) of the trace events. The trace event message is printed if any of the tags matches. If the value is empty or ALL, all the messages are printed. Available tags: - ALL - <b>UNPREDICTABLE</b> - IMP_DEF
suppress_categories	String	"Why"	""	Init time	Space-separated list of categories which should not be printed
suppress_repeated	Bool	0x1	true, false	Init time	Suppress repeated messages from similar call sites
suppress_sources	String	""	""	Init time	Space-separated list of components or events that should not be printed
trace-file	String	""	""	Init time	ArchMsgTrace output file

**4.4 ASTFplugin**

ASTFplugin is an MTI plug-in that enables Fast Models to generate trace output in Architectural Structured Trace Format (ASTF).

ASTF is a binary, compressible trace format that captures the architectural execution of each of the CPUs in a system. It supports the collection of traces from complex workloads of up to billions of instructions in length. The format was designed to achieve a balance between compactness, ease of interpretation, and strong forwards and backwards compatibility.

ASTF and associated tools have been developed to support workload tracing, workload analysis, and to drive CPU performance models.

ASTFplugin can be used in combination with [ToggleMTIPlugin](#).



**Note**

- ASTFplugin is supported on Linux hosts only.
- ASTFplugin and the ASTF specification are in development and further iterations are expected. For the status of ASTFplugin, the version of the specification it supports, and any limitations and known issues, see the Fast Models release notes.

## Additional reading

- The ASTF specification is included in the Fast Models package in the `$FVLIB_HOME/Docs/` directory.
- For answers to some common queries about ASTFplugin, see the [ASTFplugin FAQs](#).
- For best practices for preparing an ASTF trace for performance prediction of a workload see [Workload Trace Generation Best Practices](#).
- For how to use a Fast Models FVP to capture an ASTF trace, see [How to generate ASTF traces of workloads running on Fast Models](#).

### 4.4.1 ASTFplugin - parameters

Each parameter is prefixed with `TRACE.ASTFplugin`, for example:

```
TRACE.ASTFplugin.encoding-method
```

Name	Type	Default value	Allowed values	When set	Description
encoding-method	Int	0x2	0x0 - 0x2	Runtime	ASTF record encoding method. 0: Uncompressed; 1: Compressed LZMA; 2: Compressed ZLib
register-recording	Int	0x1	0x0 - 0x1	Init time	Register records and architectural states output setting. 0: Disabled; 1: Output core registers + stack pointer + NZCV
timestamp-enable	Bool	0x1	true, false	Init time	Timestamp records will become part of the output if enabled.
timestamp-period	Int	0x5	0x1 - 0x7fffffffffffffffff	Init time	This parameter sets the simulated time between two timestamps in micro-seconds.
trace-file	String	""	""	Runtime	Trace file pathname and prefix to write out to. Will be appended with component path, session number and .astf suffix.
verbosity	Int	0x2	0x0 - 0x2	Runtime	Output verbosity level. 0: FATAL; 1: ERROR; 2: WARNING

### 4.4.2 ASTFplugin usage notes

Be aware of the following points when using ASTFplugin.

- Load ASTFplugin in the same way as other plug-ins, using the syntax:

```
./isim_system <isim_params> --plugin /path/to/ASTFplugin.so <astf_plugin_params>
```

- ASTFplugin generates trace files with a `.astf` extension. During the simulation, these trace files might be incomplete. Incomplete trace files have a `.astf.part` extension and cannot be processed using the ASTF tools.
- The output trace files have a four digit enumerator field in the name. For example `FVP_Base_Cortex_A55.cluster0.cpu0.0000.astf`. This enumerator field is always present, regardless of whether ASTFplugin is used together with ToggleMTIPlugin. If ToggleMTIPlugin instructs ASTFplugin to stop and then resume, a new file is created for each CPU with each enumerator field incremented by one. However, if a CPU was not active when ToggleMTIPlugin instructed ASTFplugin to record, the respective output file is not created.
- ASTFplugin tries to register callbacks for MTI trace sources for the Scalable Vector Extension (SVE). If SVE is not enabled, ASTFplugin reports warnings to the console. You can ignore these warnings if SVE operations do not need to be recorded or if SVE is intended to be disabled.
- To improve performance, ASTFplugin is multithreaded. As the plug-in handles large streams of data, avoid using SMT or Hyper-Threading or running the threads on different sockets on a multi-socket host system. For optimal performance, we recommend you use `taskset` to restrict the model to using a specified set of N+1 host cores where N is the number of cores simulated in the model.

### 4.4.3 Additional ASTF support in Fast Models

In addition to ASTFplugin, Fast Models includes some other tools and libraries that support ASTF.

- The `trprint`, `trcheck`, `trdd`, `trimage`, and `trpidannotate` tools enable you to process the ASTF trace file. For details, see [ASTF tools](#).
- [ToggleMTIPlugin](#), installed in `$PVLIB_HOME/plugins/<OS_Compiler>/`, can be used together with ASTFplugin to limit trace generation to specific regions of interest.
- `libastf` library and header files:

**`$PVLIB_HOME/lib/<platform_compiler>/libastf.a`**

Library for reading and writing ASTF trace files. It exposes both C++ and C interfaces. For documentation, including a basic C++ example, see `$PVLIB_HOME/Docs/astf/libastf-api/libastf-api.txt`.

**`$PVLIB_HOME/include/astf/astf.h`**

Specifies the C++ and C interfaces to `libastf`. For documentation for each API function, see `$PVLIB_HOME/Docs/astf/libastf-api/<C++_function_name>.txt`.

**`$PVLIB_HOME/include/astf/astf_records.h`**

ASTF library record definitions.

- An example Python script, `$PVLIB_HOME/examples/PyIris/inst_count_trace_control.py`. It uses the `iris.debug` Python module to demonstrate using ToggleMTIPlugin to limit tracing to specific parts of the application. For usage instructions, run the script with the `-h` option. For more information, see the comments in the source file.

### Related information

[Iris Python Debug Scripting User Guide](#)

## 4.4.4 ASTF tools

The ASTF-related tools `trcheck`, `trdd`, `trimage`, `trpidannotate`, and `trprint` are installed in `$PVLIB_HOME/bin/astf_tools/`. They enable you to process the trace files that ASTFplugin outputs, for example to view them in a human-readable format.

### **trcheck**

Verifies the correctness of the trace files against the semantics defined in the ASTF format specification. For documentation, see `$PVLIB_HOME/Docs/astf/tools/trcheck.txt`.

### **trdd**

Slices, copies, and (re)compresses the trace files. It can cut pieces from a trace file or re-encode a trace file by using a different compression level. For documentation, see `$PVLIB_HOME/Docs/astf/tools/trdd.txt`.

### **trimage**

Analyses and profiles instructions and branches across multiple ASTF files. For documentation, see `$PVLIB_HOME/Docs/astf/tools/trimage.txt`.

### **trpidannotate**

Annotates Context records in the trace files to correct the PID/TID information that was collected during tracing. For documentation, see `$PVLIB_HOME/Docs/astf/tools/trpidannotate.txt`.

### **trprint**

Enables viewing and printing trace files in a human-readable format. For documentation, see `$PVLIB_HOME/Docs/astf/tools/trprint.txt`.

## 4.4.5 ASTFplugin FAQs

These FAQs answer some common queries about ASTFplugin.

### **How do I trace a workload running in a multi-core Linux environment?**

I want to trace an application's workload running in a guest multi-core Linux environment but Linux OS migrates applications between cores. What should I do?

When using the `HLT` method of toggling trace with `ToggleMTIPlugin`, ASTF tracing in Fast Models is per-core. So, you must enable the `enable_trace_special_hlt_imm16` and `trace_special_hlt_imm16` parameters for all of the cores in the simulation.

See also [How to use ToggleMTIPlugin](#).

### **What are the architectural limitations of ASTFplugin?**

ASTF v0.11 supports up to Armv9.3-A. The plug-in is not guaranteed to work for architecture versions later than that.

### **Does ASTF trace generation have a maximum trace file size?**

No, it runs until your disk space is full and is killed by your OS.

## How much trace data should I capture?

A minimum trace log size of around 100M instructions is recommended to account for cache warming. A maximum of 10B instructions is recommended for efficient post-processing and analysis.

## If I generate traces for multiple programs, how do I distinguish between them in the trace logs?

If the guest Linux kernel is configured with 'CONFIG\_PID\_IN\_CONTEXTIDR' enabled, PID information is included in the `context` section of the trace logs. For example:

```
5 context : CPU in EL0, non-secure, thread-mode PID: 29193
```



Note

PID information is only recorded if there is a PID change on a core.

## Can ASTFplugin trace process and thread IDs?

ASTFplugin supports recording PID/TID information through the `CONTEXTIDR_EL1` register, if the guest supports it. However that requires an additional pass for full PID/TID information. Is there a way to collect PID/TID information that doesn't require a re-run?

It is possible to do this with some additional post-processing. ASTFplugin traces PIDs using the `CONTEXTIDR_EL1` register and includes them in the trace, if they are available, in the first run. To fully match up the PIDs and TIDs in the trace, you then need to generate a PID-TID map from the OS and use the `trpidannotate` tool to amend the trace with the appropriate TID information.

See also [ASTF tools](#).

## Are timing controls, for example `cpi_div` and `cpi_mul`, or Timing Annotation settings changeable at runtime?

No, they are only changeable at instantiation time.

## How does the plug-in distinguish between different clusters and cores?

The plug-in has no concept of the cluster and core topology of the model. It simply queries whether each component can execute code. If it can, the plug-in attaches itself to the trace sources of the component that it needs to generate the ASTF streams. If not, it ignores that component. The resulting file names that include the clusters and cores are generated by the model and accepted by the plug-in.

## Why do files seem to be missing?

For example, in a model with two CPUs, I want to record two blocks. Files `cpu0.0000.astf` and `cpu1.0000.astf` record the first block and `cpu0.0001.astf` records the second block. Why is `cpu1.0001.astf` missing?

If a CPU is inactive while the plug-in is recording, the plug-in omits generating the associated file. Otherwise, the file would only contain the ASTF header and nothing else. So if a file is missing, it might be because that CPU was not active while the plug-in was recording.

### Why does the running counter for my files sometimes start with 0000 and sometimes with 0001?

When you request the plug-in to stop recording, it increments the running counter to ensure the next block's ASTF files have a different filename to those of the current block.

By default, the plug-in starts recording after it has initialised. However, if you request the plug-in to stop recording immediately, it increments the running counter and produces no `cpu*.0000.astf` files, because none of the CPUs had the chance to execute code. If you then request the plug-in to continue recording, the next block's stream files will have the 0001 counter value in their names.

### Why do I see messages about missing trace sources?

For example:

```
trace source SVE_LOADS/SVE_STORES not detected -> omitting registration
```

The plug-in is not feature-aware and therefore does not know if a CPU supports SVE. As a result, it always tries to register `SVE_LOADS` and `SVE_STORES` trace sources. If a CPU does not support SVE and this message appears, you can safely ignore it.

## 4.5 CDE

Custom Datapath Extension (CDE) allows you to improve performance and efficiency by adding application domain-specific features to embedded processors, while maintaining the advantages of the Arm® software ecosystem.

CDE allows you to add a customizable module inside some Cortex®-M processors. This module is driven by the pre-decoded CDE instructions and shares the same interface as the standard Arithmetic Logic Unit (ALU) of the processor.

Fast Models implements CDE using Model Trace Interface (MTI) plug-ins, with parameters to allow the plug-ins to be configured at runtime. The following Fast Models support CDE:

- [ARMCortexM33CT](#)
- [ARMCortexM52CT](#)
- [ARMCortexM55CT](#)
- [ARMCortexM85CT](#)
- [ARMAEMv8MCT](#)

The following model parameters are exposed for configuring CDE:

**has\_cde**

Controls whether CDE is enabled. If enabled, a plug-in must be provided.

**--plugin <path/to/plugin.so>**

This option can be specified multiple times, once for each CDE plug-in implementation. Alternatively, plug-ins can be loaded by setting the `FM_PLUGINS` environment variable.

**cpu.cde\_impl\_name=<plugin\_name>**

The CDE implementation name to use with this core. If multiple CDE plug-in implementations are provided, each core can be requested to use a specific plug-in by using `cpu<n>.cde_impl_name=...`

Two example plug-ins are available, CDETester and CDEConstant. They are provided as pre-built libraries and as source code, located in `$PVLIB_HOME/plugins/source/`, to help with implementing your own plug-ins.

## 4.5.1 CDETester

CDETester is a basic example plug-in that allows you to specify at runtime which CDE instructions are supported by individual coprocessors and to specify the behavior, either nop or undefined.

To specify the instructions that coprocessors support, provide a plug-in parameter of the form:

```
CDE.CDETester.cde_tester_trivial.cps_implemented_instr=0xn
```

where `0xn` represents a hexadecimal bitmask of coprocessors that implement this instruction and `instr` represents a CDE instruction name.

The full list of CDE instruction names is as follows, where `a` represents dual variants and `v` represents vector variants:

- `cx1`
- `cx2`
- `cx3`
- `cx1d`
- `cx2d`
- `cx3d`
- `vcx1`
- `vcx2`
- `vcx3`
- `vcx1v`
- `vcx2v`
- `vcx3v`

Accumulate variants are handled in the same function implementation as the non-accumulate variants.

The bitmask takes the form:

**bits [7:0]**

For non-accumulate variants.

**bits [23:16]**

For accumulate variants.

The plug-in also allows control over which coprocessors implement CDE through the `CDE.CDETester.cde_tester_trivial.cps_implemented=0xn` parameter, where `0xn` represents a hexadecimal bitmask of coprocessors that implement CDE.

An example invocation to enable `cx1` and `cx1A` (accumulate) for CP3 might be written as:

```
CDE.CDETester.cde_tester_trivial.cps_implemented=0x8 // enable CDE support for CP3
CDE.CDETester.cde_tester_trivial.cps_implemented_cx1=0x80008 // enable CX1 and CX1A
on CP3 (behaves as nop rather than undef)
```

On Linux, you can retrieve an up-to-date list of parameters from the model by using:

```
-l | grep -i cdetester
```

## 4.5.2 CDETester - parameters

Each parameter is prefixed with `CDE.CDETester`, for example:

```
CDE.CDETester.cde_tester_trivial.cps_implemented
```

Name	Type	Default value	Allowed values	When set	Description
<code>cde_tester_trivial.cps_implemented</code>	Int	0xff00ff	0x0 - 0xffffffff	Init time	Bitmask indicating coprocessors implemented.
<code>cde_tester_trivial.cps_implemented_cx1</code>	Int	0xff00ff	0x0 - 0xffffffff	Init time	Coprocessor enables CX1 instructions ([23:16] for CX1A, [7:0] for CX1)
<code>cde_tester_trivial.cps_implemented_cx1d</code>	Int	0xff00ff	0x0 - 0xffffffff	Init time	Coprocessor enables CX1D instructions ([23:16] for CX1DA, [7:0] for CX1D)
<code>cde_tester_trivial.cps_implemented_cx2</code>	Int	0xff00ff	0x0 - 0xffffffff	Init time	Coprocessor enables CX2 instructions ([23:16] for CX2A, [7:0] for CX2)
<code>cde_tester_trivial.cps_implemented_cx2d</code>	Int	0xff00ff	0x0 - 0xffffffff	Init time	Coprocessor enables CX2D instructions ([23:16] for CX2DA, [7:0] for CX2D)
<code>cde_tester_trivial.cps_implemented_cx3</code>	Int	0xff00ff	0x0 - 0xffffffff	Init time	Coprocessor enables CX3 instructions ([23:16] for CX3A, [7:0] for CX3)
<code>cde_tester_trivial.cps_implemented_cx3d</code>	Int	0xff00ff	0x0 - 0xffffffff	Init time	Coprocessor enables CX3D instructions ([23:16] for CX3DA, [7:0] for CX3D)

Name	Type	Default value	Allowed values	When set	Description
cde_tester_trivial.cps_implemented_vcx1	Int	0xff00ff	0x0 - 0xffffffff	Init time	Coprocessor enables VCX1 instructions ([23:16] for VCX1A, [7:0] for VCX1)
cde_tester_trivial.cps_implemented_vcx1v	Int	0xff00ff	0x0 - 0xffffffff	Init time	Coprocessor enables VCX1 (Vector) instructions ([23:16] for VCX1A (Vector), [7:0] for VCX1 (Vector))
cde_tester_trivial.cps_implemented_vcx2	Int	0xff00ff	0x0 - 0xffffffff	Init time	Coprocessor enables VCX2 instructions ([23:16] for VCX2A, [7:0] for VCX2)
cde_tester_trivial.cps_implemented_vcx2v	Int	0xff00ff	0x0 - 0xffffffff	Init time	Coprocessor enables VCX2 (Vector) instructions ([23:16] for VCX2A (Vector), [7:0] for VCX2 (Vector))
cde_tester_trivial.cps_implemented_vcx3	Int	0xff00ff	0x0 - 0xffffffff	Init time	Coprocessor enables VCX3 instructions ([23:16] for VCX3A, [7:0] for VCX3)
cde_tester_trivial.cps_implemented_vcx3v	Int	0xff00ff	0x0 - 0xffffffff	Init time	Coprocessor enables VCX3 (Vector) instructions ([23:16] for VCX3A (Vector), [7:0] for VCX3 (Vector))
has_cde_tester_trivial	Bool	0x1	true, false	Init time	Whether the CDETester plugin is implemented (undefs all CDE instructions)

### 4.5.3 CDEConstant

This is an example plug-in that provides an implementation for every CDE instruction variant, for example [v]cxn, [v]cxnA, and [v]cxnD. Each instruction simply performs an XOR with some arguments and a constant.

To load this plug-in, use the following parameters when launching the model:

```
--plugin path/to/plugin/CDEConstant.so -C cpu.has_cde=1 -C
cpu.cde_impl_name=CDE_CONSTANT
```

### 4.5.4 CDEConstant - parameters

Each parameter is prefixed with `CDE.CDEConstant`, for example:

```
CDE.CDEConstant.has_cde_constant
```

Name	Type	Default value	Allowed values	When set	Description
has_cde_constant	Bool	0x1	true, false	Init time	Whether the CDEConstant plugin is implemented

### 4.5.5 Implementing a CDE plug-in

Fast Models supports prototyping of custom instructions through a modular plug-in system, which uses the Model Trace Interface (MTI) framework.

Multiple CDE plug-ins can be registered with the model and each core can be instructed which plug-in behavior to use. Run-time configuration of CDE plug-ins is performed using plug-in



parameters, although plug-in developers can use alternative approaches, for example configuration files.

This guide shows how to implement a basic MTI-based CDE plug-in, using the CDETester plug-in as an example. It is intended to be a quick start to plug-in development, and does not describe details about MTI. To learn more about MTI, see [Model Trace Interface Reference Manual](#). The source code for CDETester can be found in `$PVLIB_HOME/plugins/source/CDETester/`.

A CDE plug-in performs three main tasks:

- MTI and CDE interface registration.
- Handling parameters.
- Implementing CDE instructions.

### Related information

[CDETester](#) on page 6330

#### 4.5.5.1 Prerequisites for implementing a CDE plug-in

To build the CDE plug-in examples, you need the following:

- A compiler that matches the Fast Models build you are using.
- A recent version of CMake.
- An installation of the Fast Models package and libraries.

#### 4.5.5.2 CDE plug-in registration

After a CDE plug-in has been loaded into the model through the `--plugin` argument, or the `FM_PLUGINS` environment variable, it must register itself with the CDE interface registry using the MTI framework API.

The CDE interface registry is responsible for:

- Managing all loaded CDE plug-ins.
- Passing any parsed arguments to the relevant CDE plug-in.
- Assigning CDE plug-ins to cores, as requested by the model arguments.

The CDE interface registry requires an interface name and version to be registered through MTI, as shown in `CDETester.cpp` and `CDETesterTrivialImpl.h`.

### 4.5.5.3 CDE plug-in parameters

After the plug-in has registered itself with the CDE interface registry, the model passes any parsed command-line parameters to the CDE plug-in they are associated with.

The parameters to the CDETester example plug-in allow you to specify which custom instructions result in a no operation (nop) for each coprocessor. Any instructions that are not enabled result in an Undefined Instruction exception being raised.

The plug-in handles parameters that it receives from the model in `CDETester.cpp` by passing them through the `CDETesterFactory` interface to the handler implementation in `CDETrivialImpl::consumeParameter()`, defined in `CDETesterTrivialImpl.cpp`.

#### Related information

[CDETester](#) on page 6330

### 4.5.5.4 CDE plug-in instruction handler interface

After plug-in registration and optional parameter handling, the plug-in should implement handlers for all available CDE instructions, even if it does not intend to implement custom functionality for all instruction variants.

To help you do this, Fast Models provides a utility header, `$PVLIB_HOME/include/ct/CDE/CDEInstHandlerInterface.h`. This header defines the `CDEInstHandlerInterface` class for handling CDE instruction calls, which plug-ins must inherit.

This interface declares pure virtual methods for each CDE instruction variant, for example dual and accumulator, as well as the structures containing decode information and call results. A typical signature follows this pattern:

```
CDEResult64 CDETrivialImpl::exec_cx2_d(const CX2DecodeInfo& decode_info, uint64_t
    rfd_val, uint32_t rn_val)
```

The CDETester example plug-in inherits this interface in `CDETesterBaseImpl.h` and implements the instructions in `CDETesterTrivialImpl.h` and `CDETesterTrivialImpl.cpp`.

### 4.5.5.5 CDE plug-in instruction implementations

Each CDE instruction is mapped to a single function definition, except for accumulate variants, which are handled by checking for the accumulate flag in the parameters passed to the instruction implementations.

The full list of instructions that a plug-in is expected to implement is given in [CDETester](#).

Each instruction implementation accepts as parameters:

- A structure containing decoded instruction opcode parameters, including register numbers and immediate. For example `CX1DecodeInfo`.

- The contents of registers specified in the instruction opcode.

Instruction implementations should return a result structure of varying size, for example `CDEResult32` or `CDEResult64`. This structure indicates whether the instruction is supported, and if so, the return value and the number of cycles taken for execution, which is used in trace and performance analysis.

If a plug-in needs to raise an Undefined Instruction exception for a particular instruction, it can simply return a default-initialized result structure.

For full details of the expected function declarations and parameter types, see the file `$PVLIB_HOME/include/ct/CDE/CDEInstHandlerInterface.h`.

#### 4.5.5.6 Building the CDE plug-in

During plug-in development, either modify the example `CMakeLists.txt` files provided with the CDE plug-ins to reflect any changes to the file structure, or alternatively, use your own build system.

To build the example plug-ins:

1. Run `CMake` on the root directory of the plug-in to generate the project file, for example a Makefile or Visual Studio solution.
2. Run `make`.

### 4.5.6 CDE API

Reference documentation for the CDE API.

The CDE API is defined in the following header files, which are located in `$PVLIB_HOME/include/ct/CDE/`:

- `CDEFactoryInterface.h`
- `CDEInstHandlerInterface.h`
- `CDERegistryInterface.h`

#### 4.5.6.1 CDEFactoryInterface.h

Defines the interface for obtaining an instance of a `CDEInstHandlerInterface` for a specific core.

##### 4.5.6.1.1 CDE::CDEFactoryInterface class

The factory interface for obtaining an instance of a `CDEInstHandlerInterface` for a specific core.

#### 4.5.6.1.2 CDE::CDEFactoryInterface::instantiateCDEInstHandler()

Instantiate a `CDEInstHandlerInterface` for a given core. The caller owns the result.

```
virtual std::unique_ptr<CDEInstHandlerInterface>  
instantiateCDEInstHandler(std::string component_hierarchy) = 0;
```

##### **component\_hierarchy**

String representing the hierarchy of the current component.

#### 4.5.6.1.3 CDE::CDEFactoryInterface::CDEImplName()

Return the name of the CDE implementation. A core can use this method to disambiguate multiple CDE implementations in a simulation.

```
virtual std::string CDEImplName() const = 0;
```

#### 4.5.6.1.4 CDE::CDEFactoryInterface::CDEImplDescription()

Description of the CDE implementation.

Return the description of the CDE implementation that can be instantiated.

```
virtual std::string CDEImplDescription() const = 0;
```

#### 4.5.6.1.5 CDE::CDEFactoryInterface::CDEImplProviderName()

Provider name of the CDE implementation.

Return the name of the component providing the CDE implementation, for example a plug-in. This name might be used in informative diagnostic messages.

```
virtual std::string CDEImplProviderName() const = 0;
```

### 4.5.6.2 CDEInstHandlerInterface.h

Defines the interface for executing CDE instructions.

#### 4.5.6.2.1 CDE::CDEResult32 struct

32-bit result of a CDE instruction.

##### Members

###### **instr\_not\_supported**

Whether the instruction is supported.

###### **value**

Return value of the instruction.

###### **cycles**

Number of cycles of instruction execution.

#### 4.5.6.2.2 CDE::CDEResult64 struct

64-bit result of a CDE instruction.

##### Members

###### **instr\_not\_supported**

Whether the instruction is supported.

###### **value**

Return value of the instruction.

###### **cycles**

Number of cycles of instruction execution.

#### 4.5.6.2.3 CDE::CDEResult128 struct

128-bit result of a CDE instruction.

##### Members

###### **instr\_not\_supported**

Whether the instruction is supported.

###### **value\_lo**

Low 64 bits of the return value of the instruction.

###### **value\_hi**

High 64 bits of the return value of the instruction.

###### **cycles**

Number of cycles of instruction execution.

#### 4.5.6.2.4 CDE::CX1DecodeInfo struct

Decoded information for the cx1 instruction.

##### Members

###### **accumulate**

Whether to accumulate with existing register contents.

###### **coproc**

Number of coproc.

###### **imm**

Immediate value.

###### **rd\_num**

General-purpose destination register number.

#### 4.5.6.2.5 CDE::CX2DecodeInfo struct

Decoded information for the cx2 instruction.

##### Members

###### **accumulate**

Whether to accumulate with existing register contents.

###### **coproc**

Number of coproc.

###### **imm**

Immediate value.

###### **rd\_num**

General-purpose destination register number.

###### **rn\_num**

General-purpose source register number.

#### 4.5.6.2.6 CDE::CX3DecodeInfo struct

Decoded information for the cx3 instruction.

##### Members

###### **accumulate**

Whether to accumulate with existing register contents.

###### **coproc**

Number of coproc.

**imm**

Immediate value.

**rd\_num**

General-purpose destination register number.

**rn\_num**

General-purpose source register number.

**rm\_num**

General-purpose source register number.

#### 4.5.6.2.7 [CDE::VCX1DecodeInfo struct](#)

Decoded information for the vcx1 instruction.

##### Members

**accumulate**

Whether to accumulate with existing register contents.

**coproc**

Number of coproc.

**imm**

Immediate value.

**vd\_num**

Source and destination vector register number.

#### 4.5.6.2.8 [CDE::VCX2DecodeInfo struct](#)

Decoded information for the vcx2 instruction.

##### Members

**accumulate**

Whether to accumulate with existing register contents.

**coproc**

Number of coproc.

**imm**

Immediate value.

**vd\_num**

Source and destination vector register number.

**vm\_num**

Source vector register number.

#### 4.5.6.2.9 CDE::VCX3DecodeInfo struct

Decoded information for the vcx3 instruction.

##### Members

###### **accumulate**

Whether to accumulate with existing register contents.

###### **coproc**

Number of coproc.

###### **imm**

Immediate value.

###### **vd\_num**

Source and destination vector register number.

###### **vn\_num**

Source vector register number.

###### **vm\_num**

Source vector register number.

#### 4.5.6.2.10 CDE::CDEInstHandlerInterface class

Interface for executing CDE instructions.

This class defines the following methods for executing CDE instructions:

###### **exec\_cx1()**

cx1 instruction.

###### **exec\_cx1\_d()**

cx1D instruction.

###### **exec\_cx2()**

cx1 instruction.

###### **exec\_cx2\_d()**

cx2D instruction.

###### **exec\_cx3()**

cx3 instruction.

###### **exec\_cx3\_d()**

cx3D instruction.

###### **exec\_vcx\_1\_s()**

vcx1 instruction with S register.

###### **exec\_vcx\_1\_d()**

vcx1 instruction with D register.



**exec\_vcx\_1\_q()**

vcx1 instruction with Q register.

**exec\_vcx\_2\_s()**

vcx2 instruction with S register.

**exec\_vcx\_2\_d()**

vcx2 instruction with D register.

**exec\_vcx\_2\_q()**

vcx2 instruction with Q register.

**exec\_vcx\_3\_s()**

vcx3 instruction with S register.

**exec\_vcx\_3\_d()**

vcx3 instruction with D register.

**exec\_vcx\_3\_q()**

vcx3 instruction with Q register.

**exec\_vcx\_1\_beatwise()**

vcx1 instruction for one beat. Caller handles predicated writeback.

**exec\_vcx\_2\_beatwise()**

vcx2 instruction for one beat. Caller handles predicated writeback.

**exec\_vcx\_3\_beatwise()**

vcx3 instruction for one beat. Caller handles predicated writeback.

#### 4.5.6.2.11 CDE::CDEInstHandlerInterface::getCDECoprocessorMask()

Return a bitmask indicating which coprocessor numbers this CDE implementation subsumes.

```
virtual uint8_t getCDECoprocessorMask() = 0;
```

### 4.5.6.3 CDERegistryInterface.h

Defines the interface to allow components, for instance plug-ins, to contribute CDE implementations to the simulation.

#### 4.5.6.3.1 CDE::CDERegistryInterface class

Interface to register the CDE factory.

```
class CDERegistryInterface : public eslapi::CAInterface
```

This class is the interface to register the CDE factory into the Fast Models simulation component registry.

#### 4.5.6.3.2 CDE::CDERegistryInterface::registerCDEFactory()

Register the CDE factory with the simulation.

```
virtual bool registerCDEFactory(std::ostream& error_stream, CDEFactoryInterface*  
    interface) = 0;
```

**error\_stream**

The error stream.

**interface**

The CDE factory interface used to register.

#### 4.5.6.3.3 CDE::CDERegistryInterface::unregisterCDEFactory()

Unregister the CDE factory from the simulation.

```
virtual void unregisterCDEFactory(CDEFactoryInterface* interface) = 0;
```

**interface**

The CDE factory interface used to unregister.

#### 4.5.6.3.4 CDE::CDERegistryInterface::sendToCores()

Instantiate a `CDEInstHandler` and send it to the core.

The core can then call the CDE instruction execution functions provided by the plug-in on that `CDEInstHandler`.

```
virtual bool sendToCores() = 0;
```

## 4.6 CDELoader

CDELoader is a framework to enable rapid Arm Custom Instruction (ACI) prototyping.

It was introduced in Fast Models 11.27 as an alternative to the existing Custom Datapath Extension (CDE) plug-in framework.

To simplify the task of implementing Arm custom instructions, the CDELoader framework handles the required plug-in setup. The ACI developer only needs to provide a shared object, the ACI library.

The framework has the following parts:

## CDELoader

A Fast Models plug-in designed to remove the need for MTI setup from ACI development. At runtime, CDELoader loads the shared object and executes the custom instruction implementation it provides.

CDELoader can accept multiple shared objects, allowing each core to specify which one to use. It also supports runtime configuration through an opaque string, which the ACI library can process according to its own requirements.

## API layer

A C99 API that defines the interface that the ACI library is bound by. It is located in `$PVLIB_HOME/include/ct/CDE/ACILibraryAPI.h`. It includes functions related to:

- Library creation, deletion, naming, and versioning.
- Custom instruction execution. These are the `aci_exec_*` functions.
- [Custom instruction mnemonics](#).

## ACI library

A shared object that provides implementations for the custom instructions and library functions.

As the CDELoader framework uses a C99 API, the library can be written in C or in another language that can produce ABI-compatible shared objects and can interface with C. For example, C++, Rust, Go, or Python, through C extensions.



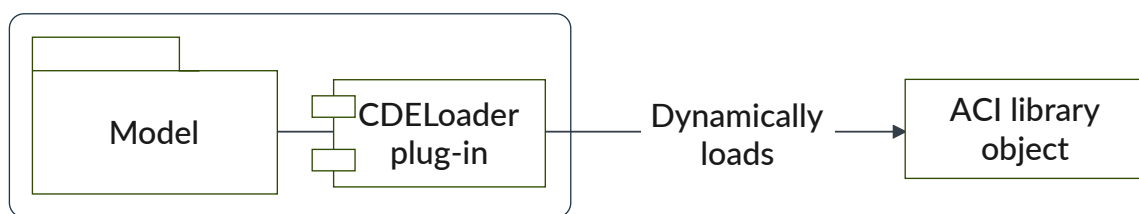
Languages other than C can require extra wrapper code to ensure proper interaction with the C99 API and to meet the shared object requirements. The implementation and compatibility details might vary depending on the language and toolchain used.

---

An example ACI library, called ACIConstant, is provided as both a pre-built library and as source code, located in `$PVLIB_HOME/plugins/source/ACIConstant/` to help you implement your own libraries. This plug-in mirrors the example [CDEConstant](#) implementation for the [CDE](#) plug-in framework.

This figure shows the framework:

**Figure 4-1: CDELoader and ACI library**



The following model parameters are exposed for using CDELoader:

**-C `cpu.has_cde=[0|1]`**

Controls whether this core enables CDE.

**-C `cpu.cde_impl_name=<aci_library_name>`**

Sets the ACI library name to use with this core. The library name is returned by `aci_get_library_name()`.

If multiple shared objects are provided, each core can request to use a specific implementation by using `cpu<n>.cde_impl_name=...`

**--plugin `<path/to/CDELoader.so>`**

Loads the CDELoader plug-in. Alternatively, set the `FM_PLUGINS` environment variable to the full path of the plug-in. For more details, see [Loading a plug-in](#).

For the CDELoader plug-in parameters, see [CDELoader - Parameters](#).

For example:

```
./isim_system \
-C cpu.has_cde=1 \
--plugin $plugin_dir/CDELoader.so \
-C cpu.cde_impl_name=ACI_LIBRARY_NAME \ # defined in aci_get_library_name()
-C CDE.CDELoader.aci_object_files="path/to/aci_library_obj_file, path/to/another/
aci_library_obj_file" \
-C CDE.CDELoader.aci_parameters="Opaque string parameter to be processed by the ACI
library"
```

## 4.6.1 CDELoader - parameters

Each parameter is prefixed with `CDE.CDELoader`, for example:

```
CDE.CDELoader.aci_object_files
```

Name	Type	Default value	Allowed values	When set	Description
<code>aci_object_files</code>	String	<code>""</code>	<code>""</code>	Init time	Comma-separated list of path to shared objects to load.
<code>aci_parameters</code>	String	<code>""</code>	<code>""</code>	Init time	Opaque string parameter to be processed by the ACI library.

## 4.6.2 ACI library implementation

Each CDE instruction is mapped to a single function definition, except for accumulate variants which are handled by checking for the accumulate flag in the parameters passed to the instruction implementations.

The following table lists the CDE instruction functions that an ACI library must implement. A `d` suffix represents a dual variant and a `v` suffix represents a vector variant.

**Table 4-6: Mapping CDE instructions to ACI library function definitions**

Instruction	ACILibraryAPI.h function
cx1	aci_exec_cx1()
cx2	aci_exec_cx2()
cx3	aci_exec_cx3()
cx1d	aci_exec_cx1_d()
cx2d	aci_exec_cx2_d()
cx3d	aci_exec_cx3_d()
vcx1 (single-register variant)	aci_exec_vx1_s()
vcx2 (single-register variant)	aci_exec_vx2_s()
vcx3 (single-register variant)	aci_exec_vx3_s()
vcx1 (double-register variant)	aci_exec_vx1_d()
vcx2 (double-register variant)	aci_exec_vx2_d()
vcx3 (double-register variant)	aci_exec_vx3_d()
vcx1v	aci_exec_vx1_beatwise()
vcx2v	aci_exec_vx2_beatwise()
vcx3v	aci_exec_vx3_beatwise()



The library must also implement some library-related functions. For full details of the library function definitions, see `$PVLIB_HOME/include/ct/CDE/ACILibraryAPI.h`.

If you do not wish to support a particular instruction or instruction variant, the function implementation can simply return `ACI_STATUS_NOT_IMPLEMENTED`.

The ACI API defines an opaque pointer, `ACIHandle`, which, for complex implementations, can be mapped to an object of a class that handles the library functions. For simple and quick implementations, you can ignore the handle pointer.

### ACI library dependencies

CDELoader dynamically loads shared objects that are standalone and do not depend on external libraries, for example compiler-specific libraries.

To resolve the external dependencies, either include them statically when compiling the shared object or ensure they exist in a path that is visible to the model. Always list all dependencies for your shared objects and check they are present before running the model.

### 4.6.3 ACIConstant example ACI library

You can find the source code for an example ACI library which implements `ACILibraryAPI.h` in `$PVLIB_HOME/plugins/source/ACIConstant/ACILibrary.cpp`.

The implementation mirrors the output of the [CDEConstant](#) plug-in.

The example is also provided as a pre-built library. To load this library, add the following parameters when launching the model:

```
--plugin $plugin_dir/CDELoader.so \  
-C cpu.has_cde=1 \  
-C cpu.cde_impl_name=ACI_CONSTANT \  
-C CDE.CDELoader.aci_object_files=$plugin_dir/ACIConstant.so
```

Alternatively, you can build the library yourself using the example Makefile provided with the ACIConstant source code. If you have problems loading the library while running the model, ensure all necessary compiler-specific libraries are visible to the model.

## 4.6.4 Custom instruction mnemonics

This is an optional feature that allows you to specify meaningful, human-readable mnemonics for your custom instructions to improve the readability of log files.

`ACILibraryAPI.h` declares a struct named `ACIMnemonics` which contains the strings used as custom mnemonics. These strings can be retrieved using the `aci_get_custom_mnemonics()` function, and they can subsequently be used by trace plug-ins loaded with the model.

Refer to the [ACIConstant example ACI library](#) source code for an example on how to define custom mnemonics.

## 4.6.5 ACI library API

Reference documentation for the ACI library API.

The ACI library API is defined in the header file `ACILibraryAPI.h`, which is located at `$PVLIB_HOME/include/ct/CDE/ACILibraryAPI.h`.

### 4.6.5.1 ACIHandle

Opaque type specifying a handle for the ACI library. This handle is passed to all the `aci_exec_*` functions.

```
typedef struct ACIHandleInstance* ACIHandle;
```

#### 4.6.5.2 aci\_get\_library\_version()

Return the library version, `ACI_API_VERSION`.

This is part of a check conducted by the model to confirm that both it and the library are operating on the same version of this API.

```
ACI_EXPORT uint16_t aci_get_library_version(void);
```

#### 4.6.5.3 aci\_get\_library\_name()

Return the library name.

The model parameter `cpu.cde_impl_name` is used to select the library. This parameter can be used to quickly change which library is used when providing multiple libraries. Each core can request to use a specific library by using `cpu<n>.cde_impl_name=...`

```
ACI_EXPORT const char* aci_get_library_name(void);
```

#### 4.6.5.4 aci\_get\_coprocessor\_mask()

Provide a bitmask that signifies the coprocessor numbers encompassed by this ACI implementation.

```
ACI_EXPORT uint8_t aci_get_coprocessor_mask(void);
```

#### 4.6.5.5 aci\_set\_param()

Sets the parameter provided to the model for the ACI library.

```
ACI_EXPORT void aci_set_param(const char* parameter);
```

##### **parameter**

A generic string that can be processed in any way that fits the library's implementation. This can be set using the model parameter `CDELoader.aci_parameters`.

#### 4.6.5.6 aci\_new()

Constructor of the ACI library.

Return a handle object which can be nullptr if there is no handle needed.

```
ACI_EXPORT ACIHandle aci_new(void);
```

#### 4.6.5.7 aci\_free()

Destructor of the ACI library.

```
ACI_EXPORT void aci_free(ACIHandle handle);
```

**handle**

Object to free, can be ignored if there was no handle allocated.

#### 4.6.5.8 ACICX1DecodeInfo struct

Decode information of cx1 instruction.

**Members****accumulate**

Whether to accumulate with existing register contents.

**coproc**

Number of coproc.

**imm**

Immediate value.

**rd\_num**

General-purpose destination register number.

#### 4.6.5.9 aci\_exec\_cx1()

Instruction CX1.

Return ACI\_STATUS\_OK on success, or ACI\_STATUS\_NOT\_IMPLEMENTED if instruction is not implemented.

```
ACI_EXPORT ACI_Status aci_exec_cx1(ACIHandle handle,
                                   const ACICX1DecodeInfo* decode_info,
                                   uint32_t rd_val,
                                   uint32_t* result);
```

**handle**

ACI handle object created by aci\_new().

**decode\_info**

Decoded fields for the CX1 instruction.

**rd\_val**

Value of the destination register.



**result**

Pointer to the value returned by the CX1 instruction.

#### 4.6.5.10 `aci_exec_cx1_d()`

Instruction CX1D.

Return `ACI_STATUS_OK` on success, or `ACI_STATUS_NOT_IMPLEMENTED` if instruction not implemented.

```
ACI_EXPORT ACI_Status aci_exec_cx1_d(ACIHandle handle,
                                   const ACICX1DecodeInfo* decode_info,
                                   uint64_t rfd_val,
                                   uint64_t* result);
```

**handle**

ACI handle object created by `aci_new()`.

**decode\_info**

Decoded fields for the CX1D instruction.

**rfd\_val**

Value of the first of the destination register pair.

**result**

Pointer to the value returned by the CX1D instruction.

#### 4.6.5.11 `ACICX2DecodeInfo` struct

Decode information of cx2 instruction.

### Members

**accumulate**

Whether to accumulate with existing register contents.

**coproc**

Number of coproc.

**imm**

Immediate value.

**rd\_num**

General-purpose destination register number.

**rn\_num**

General-purpose source register number.

#### 4.6.5.12 aci\_exec\_cx2()

Instruction CX2.

Return `ACI_STATUS_OK` on success, or `ACI_STATUS_NOT_IMPLEMENTED` if instruction not implemented.

```
ACI_EXPORT ACI_Status aci_exec_cx2(ACIHandle      handle,
                                   const ACICX2DecodeInfo* decode_info,
                                   uint32_t         rd_val,
                                   uint32_t         rn_val,
                                   uint32_t*        result);
```

**handle**

ACI handle object created by `aci_new()`.

**decode\_info**

Decoded fields for the CX2 instruction.

**rd\_val**

Value of the destination register.

**rn\_val**

Value of the source register.

**result**

Pointer to the value returned by the CX2 instruction.

#### 4.6.5.13 aci\_exec\_cx2\_d()

Instruction CX2D.

Return `ACI_STATUS_OK` on success, or `ACI_STATUS_NOT_IMPLEMENTED` if instruction not implemented.

```
ACI_EXPORT ACI_Status aci_exec_cx2_d(ACIHandle      handle,
                                      const ACICX2DecodeInfo* decode_info,
                                      uint64_t         rfd_val,
                                      uint32_t         rn_val,
                                      uint64_t*        result);
```

**handle**

ACI handle object created by `aci_new()`.

**decode\_info**

Decoded fields for the CX2D instruction.

**rfd\_val**

Value of the first of the destination register pair.

**rn\_val**

Value of the source register.

**result**

Pointer to the value returned by the CX2D instruction.

#### 4.6.5.14 ACICX3DecodeInfo struct

Decode information of cx3 instruction.

##### Members

**accumulate**

Whether to accumulate with existing register contents.

**coproc**

Number of coproc.

**imm**

Immediate value.

**rd\_num**

General-purpose destination register number.

**rn\_num**

General-purpose source register number.

**rm\_num**

General-purpose source register number.

#### 4.6.5.15 aci\_exec\_cx3()

Instruction CX3.

Return `ACI_STATUS_OK` ON SUCCESS, OR `ACI_STATUS_NOT_IMPLEMENTED` if instruction not implemented.

```
ACI_EXPORT ACI_Status aci_exec_cx3(ACIHandle handle,
                                   const ACICX3DecodeInfo* decode_info,
                                   uint32_t rd_val,
                                   uint32_t rn_val,
                                   uint32_t rm_val,
                                   uint32_t* result);
```

**handle**

ACI handle object created by `aci_new()`.

**decode\_info**

Decoded fields for the CX3 instruction.

**rd\_val**

Value of the destination register.

**rn\_val**

Value of the source register.

**rm\_val**

Value of the source register.

**result**

Pointer to the value returned by the CX3 instruction.

#### 4.6.5.16 `aci_exec_cx3_d()`

Instruction CX3D.

Return `ACI_STATUS_OK` on success, or `ACI_STATUS_NOT_IMPLEMENTED` if instruction not implemented.

```

ACI_EXPORT ACI_Status aci_exec_cx3_d(ACIHandle handle,
                                     const ACICX3DecodeInfo* decode_info,
                                     uint64_t rfd_val,
                                     uint32_t rn_val,
                                     uint32_t rm_val,
                                     uint64_t* result);

```

**handle**ACI handle object created by `aci_new()`.**decode\_info**

Decoded fields for the CX3D instruction.

**rfd\_val**

Value of the first of destination register pair.

**rn\_val**

Value of the source register.

**rm\_val**

Value of the source register.

**result**

Pointer to the value returned by the CX3D instruction.

#### 4.6.5.17 `ACIVCX1DecodeInfo` struct

Decoded information for the vcx1 instruction.

##### Members

**accumulate**

Whether to accumulate with existing register contents.

**coproc**

Number of coproc.

**imm**

Immediate value.

**vd\_num**

Source and destination vector register number.

**4.6.5.18 aci\_exec\_vcx1\_s()**

Instruction VCX1 with Single-register.

Return `ACI_STATUS_OK` on success, or `ACI_STATUS_NOT_IMPLEMENTED` if instruction not implemented.

```
ACI_EXPORT ACI_Status aci_exec_vcx1_s(ACIHandle      handle,
                                     const ACIVCX1DecodeInfo* decode_info,
                                     uint32_t          sd_val,
                                     uint32_t*          result);
```

**handle**

ACI handle object created by `aci_new()`.

**decode\_info**

Decoded fields for the VCX1 instruction.

**sd\_val**

32-bit value of the floating-point source and destination register.

**result**

Pointer to the value returned by the VCX1 instruction.

**4.6.5.19 aci\_exec\_vcx1\_d()**

Instruction VCX1 with Double-register.

Return `ACI_STATUS_OK` on success, or `ACI_STATUS_NOT_IMPLEMENTED` if instruction not implemented.

```
ACI_EXPORT ACI_Status aci_exec_vcx1_d(ACIHandle      handle,
                                     const ACIVCX1DecodeInfo* decode_info,
                                     uint64_t          dd_val,
                                     uint64_t*          result);
```

**handle**

ACI handle object created by `aci_new()`.

**decode\_info**

Decoded fields for the VCX1 instruction.

**dd\_val**

64-bit value of the floating-point source and destination register.

**result**

Pointer to the value returned by the VCX1 instruction.

#### 4.6.5.20 aci\_exec\_vcx1\_beatwise()

Instruction VCX1 for one beat. Caller handles predicated writeback.

Return `ACI_STATUS_OK` on success, or `ACI_STATUS_NOT_IMPLEMENTED` if instruction not implemented.

```

ACI_EXPORT ACI_Status aci_exec_vcx1_beatwise(ACIHandle          handle,
                                             const ACIVCX1DecodeInfo* decode_info,
                                             uint32_t          d_val,
                                             uint8_t           beat,
                                             uint8_t           elmt_mask,
                                             uint32_t*         result);

```

**handle**

ACI handle object created by `aci_new()`.

**decode\_info**

Decoded fields for the VCX1 instruction.

**d\_val**

32-bit value of the source and destination vector register.

**beat**

Beat-number to index into the Q regs.

**elmt\_mask**

Element mask.

**result**

Pointer to the value returned by the VCX1 instruction.

#### 4.6.5.21 ACIVCX2DecodeInfo struct

Decoded information for the vcx2 instruction.

**Members****accumulate**

Whether to accumulate with existing register contents.

**coproc**

Number of coproc.

**imm**

Immediate value.

**vd\_num**

Source and destination vector register number.

**vm\_num**

Source vector register number.

#### 4.6.5.22 aci\_exec\_vcx2\_s()

Instruction VCX2 with Single-register.

Return `ACI_STATUS_OK` on success, or `ACI_STATUS_NOT_IMPLEMENTED` if instruction not implemented.

```

ACI_EXPORT ACI_Status aci_exec_vcx2_s(ACIHandle          handle,
                                     const ACIVCX2DecodeInfo* decode_info,
                                     uint32_t             sd_val,
                                     uint32_t             sm_val,
                                     uint32_t*            result);

```

**handle**

ACI handle object created by `aci_new()`.

**decode\_info**

Decoded fields for the VCX2 instruction.

**sd\_val**

32-bit value of the floating-point source and destination register.

**sm\_val**

32-bit value of the floating-point source register.

**result**

Pointer to the value returned by the VCX2 instruction.

#### 4.6.5.23 aci\_exec\_vcx2\_d()

Instruction VCX2 with Double-register.

Return `ACI_STATUS_OK` on success, or `ACI_STATUS_NOT_IMPLEMENTED` if instruction not implemented.

```

ACI_EXPORT ACI_Status aci_exec_vcx2_d(ACIHandle          handle,
                                     const ACIVCX2DecodeInfo* decode_info,
                                     uint64_t             dd_val,
                                     uint64_t             dm_val,
                                     uint64_t*            result);

```

**handle**

ACI handle object created by `aci_new()`.

**decode\_info**

Decoded fields for the VCX2 instruction.

**dd\_val**

64-bit value of the floating-point source and destination register.

**dm\_val**

64-bit value of the floating-point source register.

**result**

Pointer to the value returned by the VCX2 instruction.

#### 4.6.5.24 `aci_exec_vcx2_beatwise()`

Instruction VCX2 for one beat. Caller handles predicated writeback.

Return `ACI_STATUS_OK` on success, `ACI_STATUS_NOT_IMPLEMENTED` if instruction not implemented.

```
ACI_EXPORT ACI_Status aci_exec_vcx2_beatwise(ACIHandle handle,
                                             const ACIVCX2DecodeInfo* decode_info,
                                             uint32_t d_val,
                                             uint32_t m_val,
                                             uint8_t beat,
                                             uint8_t elmt_mask,
                                             uint32_t* result);
```

**handle**

ACI handle object created by `aci_new()`.

**decode\_info**

Decoded fields for the VCX2 instruction.

**d\_val**

32-bit value of the source and destination vector register.

**m\_val**

32-bit value of the source vector register.

**beat**

Beat-number to index into the Q regs.

**elmt\_mask**

Element mask.

**result**

Pointer to the value returned by the VCX2 instruction.

#### 4.6.5.25 `ACIVCX3DecodeInfo` struct

Decoded information for the vcx3 instruction.

### Members

**accumulate**

Whether to accumulate with existing register contents.

**coproc**

Number of coproc.



**imm**

Immediate value.

**vd\_num**

Source and destination vector register number.

**vn\_num**

Source vector register number.

**vm\_num**

Source vector register number.

#### 4.6.5.26 `aci_exec_vcx3_s()`

Instruction VCX3 with Single-register.

Return `ACI_STATUS_OK` on success, or `ACI_STATUS_NOT_IMPLEMENTED` if instruction not implemented.

```
ACI_EXPORT ACI_Status aci_exec_vcx3_s(ACIHandle          handle,
                                     const ACIVCX3DecodeInfo* decode_info,
                                     uint32_t             sd_val,
                                     uint32_t             sn_val,
                                     uint32_t             sm_val,
                                     uint32_t*             result);
```

**handle**

ACI handle object created by `aci_new()`.

**decode\_info**

Decoded fields for the VCX3 instruction.

**sd\_val**

32-bit value of the floating-point source and destination register.

**sn\_val**

32-bit value of the floating-point source register.

**sm\_val**

32-bit value of the floating-point source register.

**result**

Pointer to the value returned by the VCX3 instruction.

#### 4.6.5.27 `aci_exec_vcx3_d()`

Instruction VCX3 with Double-register.

Return `ACI_STATUS_OK` on success, or `ACI_STATUS_NOT_IMPLEMENTED` if instruction not implemented.

```
ACI_EXPORT ACI_Status aci_exec_vcx3_d(ACIHandle          handle,
                                     const ACIVCX3DecodeInfo* decode_info,
```

```
uint64_t dd_val,
uint64_t dn_val,
uint64_t dm_val,
uint64_t* result);
```

**handle**

ACI handle object created by `aci_new()`.

**decode\_info**

Decoded fields for the VCX3 instruction.

**dd\_val**

64-bit value of the floating-point source and destination register.

**dn\_val**

64-bit value of the floating-point source register.

**dm\_val**

64-bit value of the floating-point source register.

**result**

Pointer to the value returned by the VCX3 instruction.

#### 4.6.5.28 `aci_exec_vcx3_beatwise()`

Instruction VCX3 for one beat. Caller handles predicated writeback.

Return `ACI_STATUS_OK` on success, or `ACI_STATUS_NOT_IMPLEMENTED` if instruction not implemented.

```
ACI_EXPORT ACI_Status aci_exec_vcx3_beatwise(ACIHandle handle,
const ACIVCX3DecodeInfo* decode_info,
uint32_t d_val,
uint32_t n_val,
uint32_t m_val,
uint8_t beat,
uint8_t elmt_mask,
uint32_t* result);
```

**handle**

ACI handle object created by `aci_new()`.

**decode\_info**

Decoded fields for the VCX3 instruction.

**d\_val**

32-bit value of the source and destination vector register.

**n\_val**

32-bit value of the source vector register.

**m\_val**

32-bit value of the source vector register.

**beat**

Beat-number to index into the Q regs.

**elmt\_mask**

Element mask.

**result**

Pointer to the value returned by the VCX3 instruction.

#### 4.6.5.29 ACIMnemonics struct

Optional custom mnemonics for the custom instructions.

##### Members

**cx1**

Instruction CX1 mnemonic.

**cx1a**

Instruction CX1 (Accumulator variant) mnemonic.

**cx1d**

Instruction CX1D mnemonic.

**cx1da**

Instruction CX1D (Accumulator variant) mnemonic.

**cx2**

Instruction CX2 mnemonic.

**cx2a**

Instruction CX2 (Accumulator variant) mnemonic.

**cx2d**

Instruction CX2D mnemonic.

**cx2da**

Instruction CX2D (Accumulator variant) mnemonic.

**cx3**

Instruction CX3 mnemonic.

**cx3a**

Instruction CX3 (Accumulator variant) mnemonic.

**cx3d**

Instruction CX3D mnemonic.

**cx3da**

Instruction CX3D (Accumulator variant) mnemonic.

**vcx1\_s**

Instruction VCX1 (Single-register variant) mnemonic.

**vcx1a\_s**

Instruction VCX1 (Single-register accumulator variant) mnemonic.

**vcx1\_d**

Instruction VCX1 (Double-register variant) mnemonic.

**vcx1a\_d**

Instruction VCX1 (Double-register accumulator variant) mnemonic.

**vcx1\_v**

Instruction VCX1 vector mnemonic.

**vcx1a\_v**

Instruction VCX1 vector (Accumulator variant) mnemonic.

**vcx2\_s**

Instruction VCX2 (Single-register variant) mnemonic.

**vcx2a\_s**

Instruction VCX2 (Single-register accumulator variant) mnemonic.

**vcx2\_d**

Instruction VCX2 (Double-register variant) mnemonic.

**vcx2a\_d**

Instruction VCX2 (Double-register accumulator variant) mnemonic.

**vcx2\_v**

Instruction VCX2 vector mnemonic.

**vcx2a\_v**

Instruction VCX2 vector (Accumulator variant) mnemonic.

**vcx3\_s**

Instruction VCX3 (Single-register variant) mnemonic.

**vcx3a\_s**

Instruction VCX3 (Single-register accumulator variant) mnemonic.

**vcx3\_d**

Instruction VCX3 (Double-register variant) mnemonic.

**vcx3a\_d**

Instruction VCX3 (Double-register accumulator variant) mnemonic.

**vcx3\_v**

Instruction VCX3 vector mnemonic.

**vcx3a\_v**

Instruction VCX3 vector (Accumulator variant) mnemonic.

## 4.7 Crypto

The `crypto` plug-in enables Armv8 and Armv9 processor models to support the Armv8.0 Cryptographic Extensions and Armv8.3 architected Pointer Authentication algorithms.

The `crypto` plug-in is available for download from [Fast Models Downloads](#).

When the plug-in is loaded:

- All Armv8 and Armv9 processors in the system implement all functionality from the Armv8.0 Cryptographic Extensions by default, although you can disable it for specific processors by setting their `CRYPTODISABLE` parameter.
- All Armv8.3 and Armv9 processors in the system implement the architected algorithms for Pointer Authentication and Generic Authentication by default. To select which processors have these algorithms enabled, use the Crypto plug-in parameters `generic_authentication_core_pattern` and `pointer_authentication_core_pattern`. These parameters accept a comma-separated list of one or more patterns to match against the instance names of cores in the system. To provide the list in a file, specify the filename, preceded by a `@` character. For example:

```
-C CRYPTO.Crypto.pointer_authentication_core_pattern=@core_patterns.conf
```

Separate multiple core patterns in the file using either commas or new lines.

AEMs, for example, [AEMvACT](#), have parameters that allow you to restrict the `crypto` plug-in features. These parameters use the same encodings as the flags within the AArch32 `ID_ISAR5` and AArch64 `ID_AA64ISAR0_EL1` system registers. You can set these parameters for a specific AEM core using this syntax:

```
-C cpu.cpu<X>.<feature_name>=<value>
```

Where `feature_name` can be one of the following:

- `crypto_aes`, with these possible values:
  - 0**  
No AES instructions are implemented
  - 1**  
The `AESE`, `AESD`, `AESMC`, and `AESIMC` instructions are implemented
  - 2**  
As 1, but in addition, the `PMULL` and `PMULL2` instructions can operate on 64-bit data values. This is the default value.
- `crypto_sha1`, with these possible values:
  - 0**  
No SHA-1 instructions are implemented

1

The SHA1C, SHA1P, SHA1M, SHA1H, SHA1SU0, and SHA1SU1 instructions are implemented. This is the default value.

- `crypto_sha256`, with these possible values:

0

No SHA-256 instructions are implemented

1

The SHA256H, SHA256H2, SHA256SU0, and SHA256SU1 instructions are implemented. This is the default value.

- `crypto_sha3`, with these possible values:

0

No Armv8.4 SHA-3 instructions are implemented. This is the default value.

1

SHA-3 instructions are implemented if Armv8.4 is enabled.

2

SHA-3 instructions are implemented.

- `crypto_sha512`, with these possible values:

0

No Armv8.4 SHA-512 instructions are implemented. This is the default value.

1

SHA-512 instructions are implemented if Armv8.4 is enabled.

2

SHA-512 instructions are implemented.

- `crypto_sm3`, with these possible values:

0

No Armv8.4 SM-3 instructions are implemented. This is the default value.

1

SM-3 instructions are implemented if Armv8.4 is enabled.

2

SM-3 instructions are implemented.

- `crypto_sm4`, with these possible values:

0

No Armv8.4 SM-4 instructions are implemented. This is the default value.

1

SM-4 instructions are implemented if Armv8.4 is enabled.

2

SM-4 instructions are implemented.

For example, to disable the AES instructions on core 0:

```
./isim_system --plugin Crypto.so -C cpu.cpu0.crypto_aes=0
```



Note

These parameters are only available for AEMs. For other Armv8-A and Armv9-A models, the behavior is fixed to the default values.

## 4.7.1 Crypto - parameters

Each parameter is prefixed with `CRYPTO.Crypto`, for example:

```
CRYPTO.Crypto.authentication_algorithm
```

Name	Type	Default value	Allowed values	When set	Description
authentication_algorithm	String	"QARMA5"	""	Init time	<p>Choice of PACAlgorithm. Valid values: "QARMA5", "QARMA3". "QARMA3" can be enabled only if the core feature has_qarma3_pac is true. "QARMA5" is unsupported in the presence of Future Architecture Technologies (FAT).</p> <p>The pattern is a comma separated list of glob-style patterns. These patterns are matched against the instance name of cores in the system. If a pattern matches then the Architected algorithm is installed on the core if that core supports ARMv8.3.</p> <p>For example: - "", install the algorithm on all supported cores - "cluster0.", install the algorithm on all cores in cluster0 - "cluster0.cpu0" install the algorithm only on cluster0.cpu0 - "cluster0.*,cluster1.cpu1" install the algorithm on all cores in cluster0 and also cluster1.cpu1 - "" (an empty string), do not install the algorithm on any core</p>
generic_authentication_core_pattern	String	"*"	""	Init time	<p>install the ARMv8.3 Architected Generic Authentication algorithm only on ARMv8.3 cores matching one of these patterns.</p> <p>The pattern is a comma separated list of glob-style patterns. These patterns are matched against the instance name of cores in the system. If a pattern matches then the Architected algorithm is installed on the core if that core supports ARMv8.3.</p> <p>For example: - "", install the algorithm on all supported cores - "cluster0.", install the algorithm on all cores in cluster0 - "cluster0.cpu0" install the algorithm only on cluster0.cpu0 - "cluster0.*,cluster1.cpu1" install the algorithm on all cores in cluster0 and also cluster1.cpu1 - "" (an empty string), do not install the algorithm on any core</p>

Name	Type	Default value	Allowed values	When set	Description
pointer_authentication_core_pattern	String	"*"	""	Init time	<p>install the ARMv8.3 Architected Pointer Authentication algorithm only on ARMv8.3 cores matching one of these patterns.</p> <p>The pattern is a comma separated list of glob-style patterns. These patterns are matched against the instance name of cores in the system. If a pattern matches then the Architected algorithm is installed on the core if that core supports ARMv8.3.</p> <p>For example: - "", install the algorithm on all supported cores - "cluster0.", install the algorithm on all cores in cluster0 - "cluster0.cpu0" install the algorithm only on cluster0.cpu0 - "cluster0.*,cluster1.cpu1" install the algorithm on all cores in cluster0 and also cluster1.cpu1 - "" (an empty string), do not install the algorithm on any core</p>
verbose	Int	0x0	0x0 - 0x1	Init time	verbosity level. 0, terse. 1, verbose

## 4.8 GenericCounter

GenericCounter is an example MTI plug-in that prints to stdout the number of occurrences of a specific trace source when the simulation terminates.

GenericCounter only counts a single trace source, which is set using the `TRACE.GenericCounter.trace-source` parameter. To count multiple trace sources, load the plug-in multiple times. In this case, each plug-in instance has a unique name which you use instead of `GenericCounter` when setting its parameters. The names are either set implicitly or explicitly. This example sets them explicitly as `counter1` and `counter2`:

```
--plugin counter1=path/to/GenericCounter.so \
--plugin counter2=path/to/GenericCounter.so \
-C TRACE.counter1.trace-source=EXCEPTION \
-C TRACE.counter2.trace-source=READ_ACCESS
```

Otherwise, each plug-in instance has an implicit name which consists of the plug-in name and a sequential suffix, for example `GenericCounter`, `GenericCounter0`, `GenericCounter1` and so on. This example uses the implicit names:

```
--plugin path/to/GenericCounter.so --plugin path/to/GenericCounter.so \
-C TRACE.GenericCounter.trace-source=EXCEPTION \
-C TRACE.GenericCounter0.trace-source=READ_ACCESS
```

The source code for this plug-in is provided in `$PVLIB_HOME/examples/MTI/GenericCounter/`.



## 4.8.1 GenericCounter - parameters

Each parameter is prefixed with `TRACE.GenericCounter`, for example:

```
TRACE.GenericCounter.print_on_event
```

Name	Type	Default value	Allowed values	When set	Description
print_on_event	String	""	""	Init time	If set, print the count information to stdout when print_on_event trace source fires. If empty, only print the count information at the end of the simulation or when the print_stats parameter is written to.
print_stats	Int	0x0	0x0 - 0x0	Runtime	On write, print count information to stdout
trace-source	String	"INST"	""	Init time	The trace source to be counted. Example: BRA_DIR

## 4.9 GenericTrace

GenericTrace is a flexible MTI plug-in that allows you to configure which events are traced by specifying a list of trace sources. Output can be printed to a file or to the terminal.

Specify one or more trace sources, separated by commas, using the `trace-sources` parameter, for example:

```
./isim_system --plugin $PVLIB_HOME/plugins/Linux64_GCC-9.3/GenericTrace.so -C  
TRACE.GenericTrace.trace-sources=EXCEPTION,EXCEPTION_RETURN
```

Alternatively, define the list of trace sources in a file. Separate the trace sources in the file using commas or semicolons. Specify the filename, preceded by a `@` character using the `trace-sources` parameter. For example:

```
-C TRACE.GenericTrace.trace-sources=@generic_trace_sources.conf
```



Tip

To see a list of the available trace sources for each component in the model that provides trace, run the model with the `ListTraceSources` plug-in. See [ListTraceSources](#) for details.

The `trace-sources` parameter is flexible:

- To specify trace sources that match a pattern, use the `*` or `?` wildcards, for example:

```
TRACE.GenericTrace.trace-sources=EXCEPTION*
```

- To collect a trace source for a specific component only, prepend the trace source with the hierarchical path of the component, for example:

```
TRACE.GenericTrace.trace-sources=FVP_Base_AEMvA.cluster0.cpu0.EXCEPTION
```

You can optionally use wildcards in either or both of the names of the trace component and the trace source, for example:

```
TRACE.GenericTrace.trace-sources=FVP_Base_AEMvA.cluster0.*.EXCEPTION*
```

- To trace specific fields in a trace source, append a field mask. For example to trace only the second field (pc) of the `INST` trace source, use:

```
TRACE.GenericTrace.trace-sources=INST=0x2
```

Or, to filter out the 8th field (`ELEMENT_SIZE`) from the `CORE_STORES` trace source, use:

```
TRACE.GenericTrace.trace-sources=CORE_STORES=0xFFFF7F
```

- If no trace sources are specified, `GenericTrace` by default traces all the instructions.

The source code for the `GenericTrace` plug-in is provided as a programming example in `$PVLIB_HOME/examples/MTI/GenericTrace/source/`.



Note

This plug-in can be used with [ToggleMTIPlugin](#).

### 4.9.1 GenericTrace - parameters

Each parameter is prefixed with `TRACE.GenericTrace`, for example:

```
TRACE.GenericTrace.collect-latest-data-only
```

Name	Type	Default value	Allowed values	When set	Description
collect-latest-data-only	Bool	0x0	true, false	Init time	Collect only latest N trace source fires and print upon destruction.
enabled	Bool	0x1	true, false	Runtime	If set to true, tracing is enabled.
flush	Bool	0x0	true, false	Runtime	If set to true, the trace file is flushed after every event. This has a performance impact but could be used to better debug crashes.
hide-fieldnames	Bool	0x0	true, false	Runtime	Do not print field names when printing trace output.

Name	Type	Default value	Allowed values	When set	Description
latest-data-size	Int	0x1	0x0 - 0x7fffffffffffffffff	Init time	Size of latest data to store in buffer for capturing only the latest trace source fires
mti_enum_wrap_value_in_quotes	Bool	0x0	true, false	Init time	Enclose MTI_ENUM values in quotes
perf-period	Int	0x0	0x0 - 0x7fffffffffffffffff	Init time	Print performance information every N instructions. 0 means disabled.
print-timestamp	Bool	0x0	true, false	Runtime	Start each trace entry with the host time.
shorten-paths	Bool	0x1	true, false	Runtime	If set to true, the component paths of trace events are shortened by removing the common prefix. The minimal, non-ambiguous path suffix remains. If all traced sources belong to the same components, no path is logged. Default is true.
simulated-timestamp	Bool	0x0	true, false	Runtime	Start each trace entry with the simulated time.
start-icount	Int	0x0	0x0 - 0x7fffffffffffffffff	Init time	Start tracing on a certain instruction count. Tracing starts up to 2048 instructions before this count.
stop-icount	Int	0x7fffffffffffffffff	0x0 - 0x7fffffffffffffffff	Init time	Stop tracing on a certain instruction count. Tracing stops up to 2048 instructions after this count.
stop_on_event	Bool	0x0	true, false	Init time	Stop the simulation when any event is triggered.
trace-file	String	""	""	Runtime	The trace file to write into. If STDERR, prints to stderr. If empty, prints to stdout.
trace-file-limit	Int	0x0	0x0 - 0x7fffffffffffffffff	Init time	The limit of the size of the output file in bytes. The simulation is stopped when this size is reached. If 0, it is unlimited.
trace-sources	String	"INST"	""	Runtime	A comma-separated list of trace sources to be traced. A component path can be prepended, with components separated by dots. Both the component path and the trace source name can contain the wildcards * and ?. A field mask as a number in hex or decimal format can be appended with =. Example: my.subsystem.core.cpu*.TRACE_SOURCE=0x08
verbose	Bool	0x0	true, false	Runtime	Print some debugging information.

## 4.9.2 GenericTrace plug-in usage example

This example shows how to use the GenericTrace plug-in to trace accesses by the graphics driver to the registers of a GPU register model.

### Procedure

1. Use the ListTraceSources plug-in to list the trace sources that the platform provides. It is located in \$PVLIB\_HOME/plugins/<OS\_compiler>/:

```
{PATH_Model} --plugin $PVLIB_HOME/plugins/Linux64_GCC-9.3/ListTraceSources.so
```

The plug-in prints the following information to the terminal:

- All components that provide trace, including the GPU model, for example:

```
Component (292) providing trace: Kits3_Subsys.css.gpu
```

- The trace sources provided by the GPU model. For example, these are some generic trace sources provided by Mali™ GPU models:

**INFO\_ReadRegister**

Access time, addresses, data, and names of the registers that were read.

**INFO\_Reset**

GPU reset data.

**INFO\_WriteRegister**

Access time, addresses, and names of the registers that were updated, and the data before and after the update.

**INFO\_Irq**

Name and state of the IRQ signal from the GPU. The name indicates the type of IRQ signal; GPU, Job Manager, or MMU. The state can be Y for Set, or N for Clear.

**WARN\_ReadToWriteOnlyRegister**

Warning messages and addresses for the write-only registers that have been read by the graphics driver.

**WARN\_WriteToReadOnlyRegister**

Warning messages and addresses for the read-only registers that have been written by the graphics driver.

**WARN\_AccessToUnimplementedRegister**

Warning messages and addresses for the invalid registers that have been accessed by the graphics driver.

2. To trace all events from the GPU model, launch the platform with the following additional options:

```
--plugin $PVLIB_HOME/plugins/Linux64_GCC-9.3/GenericTrace.so \
-C TRACE.GenericTrace.trace-sources=Kits3_Subsys.css.gpu.* \
-C TRACE.GenericTrace.enabled=1 \
-C TRACE.GenericTrace.verbose=1 \
-C TRACE.GenericTrace.print-timestamp=1 \
-C TRACE.GenericTrace.trace-file=dp-trace-generic.log
```

Where:

- `Kits3_Subsys.css.gpu` is the GPU model listed in Step 1.
- `Kits3_Subsys.css.gpu.*` means trace all the GPU-supported trace sources. Alternatively, (not shown in this example):
  - To trace one specific GPU trace source only, add it as a suffix to the GPU model. For instance, `Kits3_Subsys.css.gpu.INFO_ReadRegister`.

- To trace multiple specific trace sources, use a comma-separated list. For instance, `kits3_Subsys.css.gpu.INFO_ReadRegister`, `kits3_Subsys.css.gpu.INFO_WriteRegister`.
- The `trace-file` option specifies the log file in which to save the trace output. If it is not used, the trace results are shown on the host terminal.

## Results

The host terminal or the log file shows details about the driver-accessed registers, such as the register addresses, data, and the access time. For example:

```
HOST TIME=1557460.545195s INFO_ReadRegister: REG_OFFSET=0x0000000000000000
VALUE=0x60000000 REG_NAME="GPU_ID"
HOST TIME=1557460.545266s INFO_ReadRegister: REG_OFFSET=0x0000000000000004
VALUE=0x07130206 REG_NAME="L2_FEATURES"
HOST TIME=1557460.545279s INFO_ReadRegister: REG_OFFSET=0x0000000000000008
VALUE=0x00000000 REG_NAME="SUSPEND_SIZE"
HOST TIME=1557460.545291s INFO_ReadRegister: REG_OFFSET=0x000000000000000c
VALUE=0x000000809 REG_NAME="TILER_FEATURES"
HOST TIME=1557460.545303s INFO_ReadRegister: REG_OFFSET=0x0000000000000010
VALUE=0x00000001 REG_NAME="MEM_FEATURES"
HOST TIME=1557460.545316s INFO_ReadRegister: REG_OFFSET=0x0000000000000014
VALUE=0x000002830 REG_NAME="MMU_FEATURES"
HOST TIME=1557460.545325s INFO_ReadRegister: REG_OFFSET=0x0000000000000018
VALUE=0x000000ff REG_NAME="AS_PRESENT"
HOST TIME=1557460.545334s INFO_ReadRegister: REG_OFFSET=0x000000000000001c
VALUE=0x00000007 REG_NAME="JS_PRESENT"
HOST TIME=1557460.545345s INFO_ReadRegister: REG_OFFSET=0x00000000000000c0
VALUE=0x00000020e REG_NAME="JS0_FEATURES"
HOST TIME=1557460.545362s INFO_ReadRegister: REG_OFFSET=0x00000000000000c4
VALUE=0x0000001fe REG_NAME="JS1_FEATURES"
HOST TIME=1557460.545364s INFO_ReadRegister: REG_OFFSET=0x00000000000000c8
VALUE=0x00000007e REG_NAME="JS2_FEATURES"
HOST TIME=1515565849.690948s gpu.INFO_WriteRegister: REG_OFFSET=0x0000000000001870
VALUE=0x00000000 UPDATED VALUE=0x00000000 REG_NAME="JOB_SLOT0_JS_FLUSH_ID_NEXT"
HOST TIME=1515565849.691304s gpu.INFO_WriteRegister: REG_OFFSET=0x0000000000001860
VALUE=0x00000000 UPDATED VALUE=0x00000001 REG_NAME="JOB_SLOT0_JS_COMMAND_NEXT"
HOST TIME=1515565849.691322s gpu.INFO_Irq: IRQ_NAME="IRQ_JOB" IRQ_STATE=Y
HOST TIME=1515565849.691561s gpu.INFO_ReadRegister: REG_OFFSET=0x000000000000100c
VALUE=0x00000001 REG_NAME="JOB_IRQ_STATUS"
HOST TIME=1515565849.691643s gpu.INFO_WriteRegister: REG_OFFSET=0x0000000000001004
VALUE=0x00000000 UPDATED VALUE=0x00000001 REG_NAME="JOB_IRQ_CLEAR"
HOST TIME=1515565849.691647s gpu.INFO_Irq: IRQ_NAME="IRQ_JOB" IRQ_STATE=N
```

### 4.9.3 Mapping between SYSREG\_UPDATE trace sources and SPSR registers

For tracing updates to SPSR\_\* registers, GenericTrace maps the fields in the registers to fields in SYSREG\_UPDATE32 or SYSREG\_UPDATE64 trace sources.

The mapping is shown in the following table:

**Table 4-10: Mapping between SYSREG\_UPDATE\* trace sources and register encodings for SPSR\_\* registers**

SYSREG_UPDATE32 or SYSREG_UPDATE64 field	Register field
opc0	R
opc	M
CRn	M1

SYSREG_UPDATE32 or SYSREG_UPDATE64 field	Register field
CRm	0
opc2	0

## 4.10 ListTraceSources

ListTraceSources is an MTI plug-in that displays a complete and self-documenting list of the trace sources that a model provides, without running the model.

The plug-in prints output for each component in the model, either to stdout or to a file. For example:

```
Component (1) providing trace: FVP_Base_Cortex_A32.address_map_terminator
(PVBusMapper, 11.30.10)
=====
Component is of type "PVBusMapper"
Version is "11.30.10"
#Sources: 2

Source ArchMsg.Error.BusActiveDuringReset (Transactions recieved at the bus slave
port whilst reset was asserted)

Source ArchMsg.Warning.BusDeadlockDetected (A potential bus deadlock has been
detected)
  Field INTERMEDIARY_RESET type:MTI_BOOL size:1 (Bus mapper intermediary reset)
  Field INTERCONNECT_RESET type:MTI_BOOL size:1 (Downstream bus interconnect
reset)
  Field SLAVE_RESET type:MTI_BOOL size:1 (Downstream bus slave reset)
...
```

The source code for this plug-in is also provided as a programming example in \$PVLIB\_HOME/examples/MTI/ListTraceSources/source/.

### 4.10.1 ListTraceSources - parameters

Each parameter is prefixed with TRACE.ListTraceSources, for example:

```
TRACE.ListTraceSources.file
```

Name	Type	Default value	Allowed values	When set	Description
file	String	""	""	Init time	File to write the list of trace sources to. Default is to write to the console.
print_components_only	Bool	0x0	true, false	Init time	If true, the plug-in prints the trace component information only, not the sources or fields.

## 4.11 RunTimeParameterTest

RunTimeParameterTest is an example MTI plug-in that demonstrates how to add new string, integer, and boolean parameters at runtime.

This plug-in is provided only as source code, in `$PVLIB_HOME/examples/MTI/RunTimeParameterTest/source/`.

## 4.12 Sidechannel

The Sidechannel plug-in allows communication between the software on the host and software on the target. It is no longer used.

### 4.12.1 Sidechannel - parameters

Each parameter is prefixed with `DEBUG.Sidechannel`, for example:

```
DEBUG.Sidechannel.diagnostics
```

Name	Type	Default value	Allowed values	When set	Description
diagnostics	Int	0x0	0x0 - 0x3	Init time	Diagnostic level (0=none, 1=calls, 2=with data dump)
interceptor	String	""	""	Init time	Interceptor .SO to load
sh-diagnostics	Int	0x0	0x0 - 0x3	Init time	Diagnostic level for semihosting mechanism

## 4.13 TarmacText

TarmacText is an MTI plug-in that extracts the architectural execution trace, also known as Tarmac, of a processor. TarmacText extracts the trace in a textual form and saves it in a file.



**Note**

TarmacText is deprecated. We recommend you only use it if you specifically require the TarmacText trace format. Otherwise, use the TarmacTrace plug-in instead.

The plug-in allows you to trace multiple components simultaneously, saving the generated traces in different files.

Enable trace generation by setting the `component` parameter to the required component name or space-separated names for multiple components. For example:

```
-C TarmacText.component='component.FVP_Base_Cortex_A57.cluster0.cpu1  
component.FVP_Base_Cortex_A57.cluster0.cpu2'
```

To provide the list of components in a file, specify the filename, preceded by a @ character using the `component` parameter. For example:

```
-C TarmacText.component=@tarmac_text_components.conf
```

Separate multiple components in the file using spaces.

The default value of the `component` parameter is empty, which means the plug-in finds and traces all active processors.

Output filenames are composed of a common prefix, configurable with the `log` parameter, followed by the name of the component, and terminated with the extension `.log`. The default value of the prefix is `tarmac`.



Note

The platform name is trimmed from the component name.

---

## 4.14 TarmacTrace

`TarmacTrace` is an MTI plug-in that prints Tarmac trace to `stdout` or to a file. This section describes the format of the output.

The trace might include instructions executed, program flow, updates to registers, memory accesses made by Arm® cores in the simulation, and other information. Plug-in parameters control the amount and type of information that is traced.

`TarmacTrace` works by enabling one or more underlying MTI trace sources. Each `trace_*` parameter enables a different set of trace sources which do not overlap with each other. For example:

### **trace\_atomic**

Traces atomic transactions at the endpoint, which can be RAM, cache, or a special component, for example `AtomicOpBusFilter`. It applies to any component that can consume the atomic service request and convert it to read-modify-write. For example, a DSU can be an endpoint for atomics if the shareability of the atomic crosses the domain boundary at the DSU. It could also be any external agent.

### **trace\_loads\_stores**

Traces `CORE_LOADS` and `CORE_STORES` MTI trace sources, which are mainly generated by the PE.

### **trace\_memory**

The same as `trace_loads_stores` but kept for backward compatibility.



**trace\_mmu**

Traces the attributes of successful memory accesses. It is generated on a fresh page walk and on successfully accessing any page for the first time.

**trace\_gpt**

Traces page table walks for Granule Protection Tables for RME. No walk takes place for a hit.



Note

- For a suite of tools for analyzing and browsing TarmacTrace instruction traces, see [Arm Tarmac Trace Utilities](#).
- TarmacTrace plug-in can be used with [ToggleMTIPlugin](#).

### 4.14.1 TarmacTrace - parameters

Each parameter is prefixed with `TRACE.TarmacTrace`, for example:

```
TRACE.TarmacTrace.end-instruction-count
```

Name	Type	Default value	Allowed values	When set	Description
end-instruction-count	Int	0x0	0x0 - 0x7fffffffffffffffff	Init time	The instruction count when the tracing should be stopped. Default is to never stop tracing.
instruction-count-is-per-target	Bool	0x1	true, false	Init time	If true (default) then the start-instruction-count and end-instruction-count parameters apply to individual targets separately. If false, all components start and stop tracing at once when the first component reaches the instruction count.
loadstore-display-width	Int	0x8	0x0 - 0x40	Init time	Memory transactions can involve up to 64 bytes. For easier readability these can be broken up into multiple memory access records with a smaller number of bytes. 0 means do not break up any transaction.
quantum-size	Int	0x2710	0x1 - 0x7fffffffff	Init time	Set the default quantum size used to compute when the tracing should start and stop, in instructions. This is overridden by the <code>CORE_INFO.QUANTUM_SIZE</code> trace source field of the component, if present.
quiet	Bool	0x0	true, false	Init time	Limit output to trace information
start-instruction-count	Int	0x0	0x0 - 0x7fffffffffffffffff	Init time	The instruction count when the tracing should start. Default is to trace from the beginning.
trace-file	String	""	""	Runtime	Trace output file. The default is an empty string, which means stdout unless <code>MTI_TARMAC_LOG</code> is set. <code>STDOUT</code> means stdout. <code>STDERR</code> means stderr. Setting this parameter at runtime causes the current trace file to be flushed and closed and a new one to be opened. Writing at runtime, <code>STDOUT</code> , <code>STDERR</code> , and <code>MTI_TARMAC_LOG</code> are not supported when <code>trace-file-per-comp=1</code> .
trace-file-per-comp	Bool	0x0	true, false	Init time	Write trace to multiple files

Name	Type	Default value	Allowed values	When set	Description
trace-inst-stem	String	""	""	Init time	Base instance path to select a group of instances to trace
trace_aarch64_vfp_full_width	Bool	0x0	true, false	Init time	Trace a write to an S or D register in AArch64 as a write to the corresponding V register
trace_atomic	Bool	0x1	true, false	Init time	Trace memory update by atomic operation.
trace_branches	Bool	0x0	true, false	Init time	Trace changes of the program flow like direct or indirect branches and exception returns.
trace_bte	Bool	0x1	true, false	Init time	Trace opcode rejected by BTE.
trace_bus_accesses	Bool	0x0	true, false	Init time	Trace bus accesses by the core. This includes accesses by the caches of the core. This considerably slows down the model.
trace_cache	Bool	0x1	true, false	Init time	Trace cache fills and evictions
trace_core_registers	Bool	0x1	true, false	Init time	Trace the core registers R0-R14, the CPSR, and the SPSR registers.
trace_cp15	Bool	0x1	true, false	Init time	Trace writes to CP15 registers.
trace_dap	Bool	0x1	true, false	Init time	Trace accesses on the debug access port
trace_ete	Bool	0x1	true, false	Init time	Trace packets generated by the ETE.
trace_events	Bool	0x1	true, false	Init time	Trace events, for example exceptions and mode changes.
trace_exception_reasons	Bool	0x1	true, false	Init time	Trace INFO_EXCEPTION_REASONS (M-class only so far)
trace_generic_events	Bool	0x0	true, false	Init time	Trace generic events.
trace_gic	Bool	0x1	true, false	Init time	Trace GIC register writes and updates.
trace_gic_reads	Bool	0x0	true, false	Init time	Trace GIC register reads.
trace_gic_signal_changes	Bool	0x0	true, false	Init time	Trace GIC FIQ/IRQ Signal Changes
trace_gic_state_change	Bool	0x0	true, false	Init time	Trace GIC interrupt state changes
trace_gic_table_walks	Bool	0x0	true, false	Init time	Trace GIC table walks.
trace_gicv3	Bool	0x1	true, false	Init time	Trace GICv3 memory mapped accesses
trace_gicv3_comms	Int	0x0	0x0 - 0x7	Init time	Trace GICv3 communications between cores and distributor. Bitfield; 1 = trace CPU; 2 = trace RDO; 4 = trace internal
trace_gicv3_its	Bool	0x0	true, false	Init time	Trace GICv3 ITS command execution
trace_gicv3_reads	Bool	0x0	true, false	Init time	Trace GICv3 memory mapped reads
trace_gicv5_stream_comm	Bool	0x0	true, false	Init time	Trace GICv5<->CPUIF stream commands
trace_gpt	Bool	0x1	true, false	Init time	Trace packets generated by the GPT.
trace_hacdbbs	Bool	0x1	true, false	Init time	Trace packets generated by the HACDBS.

Name	Type	Default value	Allowed values	When set	Description
trace_hdbss	Bool	0x1	true, false	Init time	Trace packets generated by the HDBSS.
trace_instructions	Bool	0x1	true, false	Init time	Trace instructions
trace_loads_store_memtype	Bool	0x0	true, false	Init time	Show memory type information for core loads and stores
trace_loads_stores	Bool	0x1	true, false	Init time	Trace loads and stores that are triggered by instructions. These might go into the memory subsystem, into a cache, or into a TCM. This considerably slows down the model.
trace_mask_s_regs	Bool	0x0	true, false	Init time	Represent non-updated bytes as --- in S-registers trace.
trace_memory	Bool	0x0	true, false	Init time	Trace memory accesses just outside the core.
trace_mlb	Bool	0x1	true, false	Init time	Trace packets generated by the MPAM MLB fetching.
trace_mmu	Bool	0x1	true, false	Init time	Trace mmu tablewalks and associated information.
trace_mpu_events	Bool	0x0	true, false	Init time	Trace MPU events.
trace_plb	Bool	0x1	true, false	Init time	Trace information from PLB fill and evict events
trace_slpoe2	Bool	0x0	true, false	Init time	Trace information from S1POE2 table fetches
trace_spe	Bool	0x1	true, false	Init time	Trace SPE data written to memory.
trace_tag_loads_stores	Bool	0x1	true, false	Init time	Trace tag loads and stores that are triggered by MTE instructions. These might go into the memory subsystem, into a cache, or into a TCM. This considerably slows down the model.
trace_vfp	Bool	0x1	true, false	Init time	Trace the VFP and Neon registers, including FPSCR and FPEXC.
trace_virtual_tag	Bool	0x0	true, false	Runtime	Trace virtual tag that are triggered by VMTE instructions.
unbuffered	Bool	0x0	true, false	Init time	Trace events as they arrive and flush each fwrite. Prints IT even when IS should be printed.
updated-registers	Bool	0x0	true, false	Init time	Trace the updated value of registers rather than the written value
use_inst_end_for_inst_counter	Bool	0x0	true, false	Init time	When using the instruction count as the timestamp, if true, increase the instruction count at INST_END instead of INST. When using the simulation time as the timestamp, this parameter has no effect
use_instr_cnt_as_timestamp	Bool	0x1	true, false	Init time	Use the instruction count as the timestamp instead of the simulation time

## 4.14.2 TarmacTrace file format

This topic describes conventions used in the syntax definitions for each trace type.

In the syntax definition for each trace type:

**X|Y**

Indicates a choice between X and Y.

**{X}**

Indicates that X is optional or configuration-dependent.

This is the common address definition that is used in the trace command syntax:

```
<vaddr>{ : <paddr><pas> }
```

where:

**<vaddr>**

is the virtual address in hexadecimal format. See the note after this list.

**<paddr>**

is the physical address of the instruction in hexadecimal. See the note after this list. <paddr> is only present if it is different to <vaddr>.

**<pas>**

is the address space of the physical address. Either **\_NS** for Non-secure PAS, **\_RT** for Root PAS, **\_RL** for Realm PAS, or not present for Secure PAS.



**Note**

For 64-bit addresses, the value is written as either:

- 8 hex digits, if the value can be represented in 32 bits.
- 16 hex digits otherwise.

This is the virtual regime definition that is used in the trace command syntax:

```
0x<vbase>{ _NS } <el>{ vmid=<vmid> }{ , nG asid=<asid> }
```

where:

**0x<vbase>**

is the virtual address in hexadecimal format.

**\_NS**

if present, specifies that the address is Non-secure. If not present, the address is Secure.

**<el>**

is the translation regime that owns the mapping. One of:

**EL1\_ns**

Non-secure EL1&0 regime

**EL2\_ns**

Non-secure EL2 regime

**EL1\_s**

Secure EL1&0 regime

**EL3\_s**

Secure EL3 regime

**EL1\_r1**

Realm EL1&O regime if FEAT\_RME is implemented

**EL2\_rt**

Root EL2 regime if FEAT\_RME is implemented

**<vmid>**

if present, is the VMID for Non-secure, non-hyp regimes.

**nG**

if present, specifies that the virtual regime is non-global.

**<asid>**

if present, is the ASID, for non-global regimes.

### 4.14.3 Tarmac Trace output example

This example output from the Tarmac Trace plug-in shows various types of trace output, including instruction, memory access, register, translation table walk, and TLB traces.

```
...
89 clk IT (89) 80000164 f9000820 O EL1h_n : STR x0,[x1,#0x10]
89 clk MW8 80002010:000080002010_NS 00000000 80000705
90 clk IT (90) 80000168 d28080a0 O EL1h_n : MOV x0,#0x405
90 clk R X0 00000000000000405
...
98 clk IT (98) 80000188 d5181000 O EL1h_n : MSR SCTLR_EL1,x0
98 clk R SCTLR_EL1 00000000:00001005
98 clk TTW ITLB LPAE 1:1 000080002010 0000000080000705 : BLOCK ATTRIDX=1 NS=0 AP=0
SH=3 AF=1 nG=0 16E=0 PXN=0 XN=0 ADDR=0x0000000080000000
98 clk TLB FILL cpu0.UTLB 1G 0x80000000 NS EL1_n vmid=0:0x0080000000 NS Normal
InnerShareable Inner=WriteBackWriteAllocate Outer=WriteBackWriteAllocate xn=0 pxn=0
ContiguousHint=0 xs=0
...
```

This trace shows three instructions:

- The first instruction is an `STR`, which stores the 64-bit value from register `x0` to the address in `x1 + 10` byte offset:

```
89 clk IT (89) 80000164 f9000820 O EL1h_n : STR x0,[x1,#0x10]
89 clk MW8 80002010:000080002010_NS 00000000_80000705
```

In more detail:

- `IT` is a label that indicates the type of trace event described by the line. `IT` means instruction taken. To interpret the values in this line, see [Instruction trace](#). For example:
  - `89` means this is the 89th instruction.
  - `clk` means that the preceding number is an instruction count. If `ps` was displayed here instead, this would indicate the first value is a timestamp.
  - `0x80000164` is the address from which the instruction was fetched.
  - `0xf9000820` is the 32-bit opcode of the instruction.

- o indicates the CPU execution state, in this case AArch64.
  - EL1h\_n indicates the current Exception level and Security state.
  - The rest of the line following the colon is the assembly language representation of the instruction.
- mw8 indicates an 8-byte memory write. To interpret the values in this line, see [Processor memory access trace](#). For example:
  - 0x80002010 is the virtual address to which the data was written. The value after the colon is the corresponding physical address. In this example, they are the same. The \_ns suffix indicates that it is Non-secure memory.
  - 0x00000000\_80000705 is the value of the data written. Each group of 8 digits is separated using an underscore.
- The second instruction is a `mov`, which moves the value 0x405 into register x0:

```
90 clk IT (90) 80000168 d28080a0 O EL1h_n : MOV x0,#0x405
90 clk R X0 00000000000000405
```

- r indicates a register trace. To interpret the values in this line, see [Register trace](#). For example:
  - x0 is the name of the register being written to.
  - 0x405 is the new value of x0, which is the value that was moved by the `mov` instruction.
- The third instruction is an `msr`, which writes the value 0x1005 from register x0 to System register `SCTLR_EL1`.

Writing to bit 0 of `SCTLR_EL1` enables the MMU, so that all subsequent memory accesses will be done through the MMU.

The next memory access following this instruction is an instruction fetch (not shown), so a Translation Table Walk (TTW) is required to find its address.

Following the TTW, the Translation Lookaside Buffer (TLB) is updated with the new entry that caches some of the values resulting from the TTW, for example region size, base address, cachability and sharability. This appears as a TLB trace:

```
98 clk IT (98) 80000188 d5181000 O EL1h_n : MSR SCTLR_EL1,x0
98 clk R SCTLR_EL1 00000000:00001005
98 clk TTW ITLB LPAE 1:1 000080002010 0000000080000705 : BLOCK ATTRIDX=1 NS=0
AP=0 SH=3 AF=1 nG=0 16E=0 PXN=0 XN=0 ADDR=0x0000000080000000
98 clk TLB FILL cpu0.UTLB 1G 0x80000000_NS EL1_n vmid=0:0x0080000000_NS Normal
InnerShareable Inner=WriteBackWriteAllocate Outer=WriteBackWriteAllocate xn=0
pxn=0 ContiguousHint=0 xs=0
```

- TTW indicates a translation table walk trace. To interpret the values in this line, see [Translation table walk trace](#). For example:
  - ITLB means instruction TLB.
  - LPAE means Large Physical Address Extension (LPAE)-format translation table entries.
  - 1:1 means Walk stage 1, Walk level 1.

- `0x000080002010` is the page base address and the page attributes.
- `0x0000000080000705` is the raw translation table entry.
- Following the colon is the parsed result. In this case, the LPAE region descriptor.
- `TLB` indicates a TLB trace. To interpret the values in this line, see [TLB trace](#). For example:
  - `FILL` means a TLB fill operation.
  - `cpu0.UTLB` means the operation is taking place on a Unified TLB, which is shared for I-side and D-side accesses.
  - `1G` means the TLB entry is for a 1GB page.
  - `80000000_ns` means the entry has a page base address of `0x80000000` and is Non-Secure.
  - `EL1_n` means the entry is for EL1.
  - `vmid=0:0x0080000000_ns` means the entry is tagged with a specific VMID.
  - `Normal InnerShareable` means the entry is tagged as Normal Inner-Sharable.
  - `Inner` and `outer` are the inner and outer cache attributes for this entry.
  - `xn=0` means the entry is tagged NOT Execute Never.
  - `pxn=0` means the entry is tagged NOT Privileged Execute Never.
  - `ContiguousHint=0` means the entry is not tagged as part of a set of contiguous entries that can be cached as one entry.
  - `xs=0` means the page for this TLB entry is NOT XS, indicating either lack of support for FEAT\_XS in the core, or it is non-XS memory.

#### 4.14.4 Branch Target Exception event trace

Traces Branch Target Exception events if FEAT\_BTI is implemented.

##### Output syntax

```
<time> <scale> <cpu> BTE <pc> <opcode> <btype>
```

##### Trace field descriptions

###### <time>

Either the simulation time timestamp or the instruction count, depending on the value of the `TRACE.TarmacTrace.use_instr_cnt_as_timestamp` parameter.

###### <scale>

Unit for `<time>`. A value of `ps` means simulation time, `clk` means instruction count.

###### <cpu>

Processor that generated the event.

###### <pc>

The PC, hexadecimal.

**<opcode>**

Opcode of the instruction at the PC, hexadecimal.

**<btype>**

The value of the `PSTATE.BTYPE` field, binary.

**Example**

```
50525 clk cpu0 BTE 000004000004 d503241f 10
```

## 4.14.5 Cache maintenance trace

Traces all cache maintenance operations that the processor initiates.

**Output syntax**

```
<time> <scale> <cpu> CACHE MAINTENANCE <side> <operation> <scope> <data> {<pagesize>  
<memtype>}
```

**Trace field descriptions****<time>**

Either the simulation time timestamp or the instruction count, depending on the value of the `TRACE.TarmacTrace.use_instr_cnt_as_timestamp` parameter.

**<scale>**

Unit for <time>. A value of `ps` means simulation time, `clk` means instruction count.

**<cpu>**

Processor, or other component, that gave the instruction.

**<side>**

Data or instruction cache.

**<operation>**

Clean, invalidate, or both.

**<scope>**

By MVA or set/way, to Point of Coherency or Point of Unification, Inner Sharable or not.

**<data>**

Data that is associated with the operation. If the operation is by MVA, formatted according to the common address definition, see [TarmacTrace file format](#), otherwise use raw hexadecimal.

**<pagesize>**

If the operation is by MVA, this element is the size of the memory region that is described by the TLB entry that contains the MVA.

**<memtype>**

If the operation is by MVA, this element is the type of memory in the TLB entry that contains the MVA.



## 4.14.6 Cache content trace

Traces the movement of data into and out of the cache.

### Output syntax

```
<time> <scale> <cpu> CACHE <id> LINE <line> <operation> 0x<paddr><ns>
```

### Trace field descriptions

#### <time>

Either the simulation time timestamp or the instruction count, depending on the value of the `TRACE.TarmacTrace.use_instr_cnt_as_timestamp` parameter.

#### <scale>

Unit for <time>. A value of `ps` means simulation time, `clk` means instruction count.

#### <cpu>

Processor, or other component, that gave the instruction.

#### <id>

Level and side, or system identifier, of the cache.

#### <line>

Identifier of this line uniquely within this cache, expressed in hexadecimal.

#### <operation>

Notification for this cache line. One of the following options:

##### **ALLOC**

(Processor caches) Line contains new read data.

##### **INVAL**

(Processor caches) Line contains no data.

##### **DIRTY**

(Processor caches) Line contains new write data.

##### **CLEAN**

(Processor caches) Write data is written back, still valid for reads.

##### **FILL**

(System caches) Line is filled.

##### **EVICT**

(System caches) Line is evicted due to space pressure.

##### **CLEAN**

(System caches) Line is cleaned due to maintenance operation.

##### **INVAL**

(System caches) Line is invalidated due to maintenance operation.

**<addr>**

Cache line physical address in hexadecimal.

**<ns>**

Cacheline security. Blank for Secure regime, or `_ns` for Non-secure regime.

## 4.14.7 ETE write trace

ETE packet write trace.

### Output syntax

```
<time> <scale> ETE <data> # <packet_type>
```

### Trace field descriptions

**<time>**

Either the simulation time timestamp or the instruction count, depending on the value of the `TRACE.TarmacTrace.use_instr_cnt_as_timestamp` parameter.

**<scale>**

Unit for `<time>`. A value of `ps` means simulation time, `clk` means instruction count.

**<data>**

Packet data.

**<packet\_type>**

Description of the packet type. The list of packet types is specified in the [Arm Architecture Reference Manual for A-profile architecture](#) D5.2 Summary list of ETE packets.

### Examples

- 2046 clk ETE 0x800000000000000000000000 # Alignment synchronisation Packet
- 2046 clk ETE 0x0001 # Trace Info Packet
- 2046 clk ETE 0x04 # Trace On Packet
- 2046 clk ETE 0x310000496a82 # Target address with 32 bit address IS0 with context Packet
- 2046 clk ETE 0xd4 # Atom Packet Format 6
- 2047 clk ETE 0x4a91950d06 # Exception with short address IS0 Packet

## 4.14.8 Event trace

Traces exceptions, interrupts, and exception returns. In AArch64, it also traces changes to the SPSEL and to the current exception level, by generating a `CoreEvent_ModeChange`.

### Output syntax

```
<time> <scale> {<cpu>} E <addr> <mode> 00000019 CoreEvent_ModeChange
```

```
<time> <scale> {<cpu>} E <addr> {<swi_num>} <event_number> CoreEvent_<event_name>
```

The following event types have no parameters, or have different syntax from that shown above:

**E PMU\_Snapshot**

PMU snapshot capture

**E WFI\_END**

Emitted when exiting WFI

**E WFE\_END**

Emitted when exiting WFE

**E PMUOverflow\_INSTRUCTION\_COUNTER**

PMU instruction counter overflow

**E PMUOverflow\_COUNTER <counter>\_EVENT <event>**

PMU counter overflow, with counter index and event identifier

**E simulation\_stopped**

Simulation has stopped running

**E simulation\_breakpoint\_hit**

Simulation has hit a breakpoint

**E DebugEvent\_<name> <value>**

Debug event with name and value

**E DebugEvent\_#<numeric\_id> <value>**

Debug event with numeric id and value, if event has no name

## Trace field descriptions

**<time>**

Either the simulation time timestamp or the instruction count, depending on the value of the `TRACE.TarmacTrace.use_instr_cnt_as_timestamp` parameter.

**<scale>**

Unit for <time>. `ps` means simulation time, `clk` means instruction count.

**<cpu>**

Processor, or other component, that gave the instruction.

**<addr>**

Address associated with the event, formatted according to the common address definition, see [TarmacTrace file format](#).

**<mode>**

For mode change events only, the new mode being entered.

**<swi\_num>**

For CoreEvent\_SWI, the SWI number.

**<event\_number>**

Event number. Possible values are listed in the following table.

**<event\_name>**

Event name. Possible values are listed in the following table.

In the following table, the CoreEvent\_CURRENT\_\* and CoreEvent\_LOWER\_\* events cover all the ways in which exception entry can happen in AArch64 state. For example, CoreEvent\_CURRENT\_SPx\_SYNC corresponds to a synchronous exception taken from Current Exception level with SP\_ELx, x>0.

CoreEvent\_LOWER\_64\_IRQ corresponds to an IRQ or vIRQ taken from Lower Exception level, where the implemented level immediately lower than the target level is using AArch64.

**Table 4-14: Supported values for <event\_number> and <event\_name>.**

Event number	Event name
0x00000001	CoreEvent_Reset
0x00000002	CoreEvent_UndefinedInstr
0x00000003	CoreEvent_SWI
0x00000004	CoreEvent_PrefetchAbort
0x00000005	CoreEvent_DataAbort
0x00000007	CoreEvent_IRQ
0x00000008	CoreEvent_FIQ
0x0000000E	CoreEvent_ImpDataAbort
0x00000019	CoreEvent_ModeChange
0x00000080	CoreEvent_CURRENT_SP0_SYNC
0x00000081	CoreEvent_CURRENT_SP0_IRQ
0x00000082	CoreEvent_CURRENT_SP0_FIQ
0x00000083	CoreEvent_CURRENT_SP0_ABORT
0x00000084	CoreEvent_CURRENT_SPx_SYNC
0x00000085	CoreEvent_CURRENT_SPx_IRQ
0x00000086	CoreEvent_CURRENT_SPx_FIQ
0x00000087	CoreEvent_CURRENT_SPx_ABORT
0x00000088	CoreEvent_LOWER_64_SYNC
0x00000089	CoreEvent_LOWER_64_IRQ
0x0000008A	CoreEvent_LOWER_64_FIQ
0x0000008B	CoreEvent_LOWER_64_ABORT
0x0000008C	CoreEvent_LOWER_32_SYNC
0x0000008D	CoreEvent_LOWER_32_IRQ
0x0000008E	CoreEvent_LOWER_32_FIQ
0x0000008F	CoreEvent_LOWER_32_ABORT

## Examples

- 3128 clk cluster0.cpu0 E 80200000:000080200000 EL3h 00000019 CoreEvent\_ModeChange

- 3128 clk cluster0.cpu0 E 80200000:000080200000 0000008b CoreEvent\_LOWER\_64\_ABORT
- 920000 ps E 00008128:000000008128\_NS 00000083 CoreEvent\_CURRENT\_SP0\_ABORT
- 30000 ps cpu2 E 0000000c 00dfdfdf00000003 CoreEvent\_SWI
- 0 clk cpu0 E DebugEvent\_SPIDEN 00000001

### 4.14.9 Granule protection table walk trace

Traces Granule Protection Table (GPT) walks. These events are triggered by GPT lookups.

#### Output syntax

```
<time> <scale> {<cpu>} GPTW <ISIDE|DSIDE> L<level> <address> <descaddr> : <data>
<result> {FAULT}
```

See the note later in this topic for the output syntax when a second descriptor is output for GPT descriptor validation.

#### Trace field descriptions

##### <time>

Either the simulation time timestamp or the instruction count, depending on the value of the `TRACE.TarmacTrace.use_instr_cnt_as_timestamp` parameter.

##### <scale>

Unit for <time>. `ps` means simulation time, `clk` means instruction count.

##### <cpu>

Processor, or other component, that gave the instruction.

##### <ISIDE|DSIDE>

Instruction or data TLB.

##### <level>

GPT fetch level, either 0 or 1.

##### <address>

Physical address of the lookup.

##### <descaddr>

Physical address of the GPT descriptor.

##### <data>

GPT data that was read.

##### <result>

Parsed result. One of the following values:

**BLOCK TYPE=0x01 GPI=<gpi> PGS=<pgs>**

GPT descriptor is a Block descriptor. <gpi> is the GPI value of the fetched GPT entry. <pgs> is the physical granule size.

**CONTIGUOUS TYPE=0x01 GPI=<gpi> CRS=<crs>**

GPT descriptor is a Contiguous descriptor. <gpi> is the GPI value of the fetched GPT entry. <crs> is the contiguous region size.

**TABLE TYPE=0x03 ADDR=<addr>**

GPT descriptor is a Table descriptor. <addr> is the next-level table address in hexadecimal.

**GRANULE TYPE=0x0f GPI=<gpi> PGS=<pgs>**

GPT descriptor is a Granule descriptor. <gpi> is the GPI value of the fetched GPT entry. <pgs> is the physical granule size.

**INVALID**

GPT entry is invalid.

**FAULT**

Appended to a decoded BLOCK or CONTIGUOUS descriptor when the access faults.

If Granular Data Isolation (GDI) is enabled and a L1 Granule descriptor is read, TarmacTrace might output a second descriptor for verification. The ordering of the 2 descriptors depends on whether the descriptor used for verification is 16-byte aligned:

- If it is 16-byte aligned, the output syntax is:

```
<time> <scale> <cpu> GPTW <ISIDE|DSIDE> L<level> <address>
<descaddr> : <data> (<desc_for_verification>) <result>
```

- If it is not 16-byte aligned, it is output before the descriptor that is used in the GPT walk. In this case, the output syntax is:

```
<time> <scale> <cpu> GPTW <ISIDE|DSIDE> L<level> <address>
<descaddr> : (<desc_for_verification>) <data> <result>
```

where:

**<data>**

The descriptor that is used in the GPT walk.

**<desc\_for\_verification>**

The descriptor that is used for verification. This field appears in parentheses.

**Example**

```
837897720000 ps cpu0 GPTW DSIDE L0 001000000000 000000000000e200 : 0000000000000041
(00000000000000001) BLOCK TYPE=0x01 GPI=0x04 PGS=4KB FAULT
```

## 4.14.10 Instruction trace

Generates one record for every instruction started.

### Output syntax

```
<time> <scale> <cpu> <IT|IS> (<inst_id>) <addr> <opcode> <A|T|X|O>
<mode>_<security> : <disasm>
```

### Trace field descriptions

#### <time>

Either the simulation time timestamp or the instruction count, depending on the value of the `TRACE.TarmacTrace.use_instr_cnt_as_timestamp` parameter.

#### <scale>

Unit for <time>. `ps` means simulation time, `clk` means instruction count.

#### <cpu>

Processor, or other component, that gave the instruction.

#### <IT|IS>

##### IT

Instruction passed the condition code (taken).

##### IS

Instruction failed the condition code (skipped).

#### <inst\_id>

Tick count of this processor, which is equivalent to the number of instructions that are executed, except for certain instructions like `WFI/WFE` (decimal value).

#### <addr>

Fetch source address for this instruction. Formatted according to the common address definition, see [TarmacTrace file format](#).

#### <opcode>

16-bit or 32-bit hexadecimal opcode of the instruction.

#### <A|T|X|O>

Instruction set:

##### A

A32

##### T

T32

##### X

T32EE

##### O

A64

**<mode>**

Processor execution mode.

AArch32 modes:

- svc
- irq
- fiq
- usr
- mon
- sys
- abt
- und
- hyp

AArch64 modes:

- EL3h
- EL3t
- EL2h
- EL2t
- EL1h
- EL1t
- EL0t

M-profile modes:

- thread
- handler
- lockup. This indicates the core has entered the LOCKUP state.

**<security>**

Processor security state (*s* or *ns*).

**<disasm>**

Disassembly of the instruction.



### 4.14.11 Memory bus trace

Traces transactions that are initiated through the memory bus master port of the processor. These accesses use physical addresses.

#### Output syntax

```
<time> <scale> {<cpu>} B<rw><sz><fd><lk><p><s> I<wrcbs> O<wrcbs> <manager_id> <addr>
<data>
```

#### Trace field descriptions

##### <time>

Either the simulation time timestamp or the instruction count, depending on the value of the `TRACE.TarmacTrace.use_instr_cnt_as_timestamp` parameter.

##### <scale>

Unit for <time>. `ps` means simulation time, `clk` means instruction count.

##### <cpu>

Processor, or other component, that gave the instruction.

##### <rw>

**R**

Read access.

**W**

Write access.

##### <sz>

Size of the data transfer in bytes.

##### <fd>

**I**

Opcode fetch.

**D**

Data load/store or an MMU access.

##### <lk>

**L**

Locked access.

**X**

Exclusive access.

**\_**, **underscore**

Normal access.

**<p>****P**

Privileged access.

**\_, underscore**

Normal access.

**<s>****S**

Secure access.

**N**

Non-secure access.

**RL**

Realm access, when FEAT\_RME is implemented.

**RT**

Root access, when FEAT\_RME is implemented.

**I<wrcbs>**

Inner cache attributes. See O&lt;wrcbs&gt;.

**O<wrcbs>**

Outer cache attributes:

**<w>****W**

Allocate on write.

**\_, underscore**

No allocate on write.

**<r>****R**

Allocate on read.

**\_, underscore**

No allocate on read.

**<c>****C**

Cacheable access.

**\_, underscore**

Non-cacheable access.

**<b>****B**

Bufferable access.

**\_, underscore**

Non-bufferable access.

**<s>****S**

Shareability access.

**\_, underscore**

Non-shareability access.

**<manager\_id>**

Manager ID of the transaction.

**<addr>**

Physical address that is used to access memory, in hexadecimal format.

**<data>**

Hexadecimal value of data transferred. The data padding is according to the size of the transfer. Byte ordering is from lowest to highest byte. This ordering means that for accesses in little endian mode, the data occurs mirrored compared to the register/memory access records.

#### 4.14.12 POE2 fill and evict trace

Traces POE2 Permission Lookaside Buffer (PLB) fill and evict events.

##### Output syntax

```
<time> <scale> {<cpu>} PLB FILL IRT<P|U> <regime> ASID=<asid> VMID=<vmid>
  TIndex=<tindex> POIndex=<poindex>

<time> <scale> {<cpu>} PLB EVICT IRT<P|U> <regime> ASID=<asid> VMID=<vmid>
  TIndex=<tindex> POIndex=<poindex>

<time> <scale> {<cpu>} PLB FILL DPOT<range> <regime> ASID=<asid> VMID=<vmid>
  POTIndex=<potindex> POIndex=<poindex>

<time> <scale> {<cpu>} PLB EVICT DPOT<range> <regime> ASID=<asid> VMID=<vmid>
  POTIndex=<potindex> POIndex=<poindex>

<time> <scale> {<cpu>} PLB FILL TTT<P|U> <regime> ASID=<asid> VMID=<vmid>
  POTIndex=<potindex> TIndex=<tindex>

<time> <scale> {<cpu>} PLB EVICT TTT<P|U> <regime> ASID=<asid> VMID=<vmid>
  POTIndex=<potindex> TIndex=<tindex>
```

## Trace field descriptions

### <time>

Either the simulation time timestamp or the instruction count, depending on the value of the `TRACE.TarmacTrace.use_instr_cnt_as_timestamp` parameter.

### <scale>

Unit for <time>. `ps` means simulation time, `clk` means instruction count.

### <cpu>

Processor that caused the PLB fill or evict.

### <P|U>

Whether the access is to the privileged (P) or unprivileged (U) table.

### <regime>

Translation regime of this PLB entry, see the translation regime definition in [TarmacTrace file format](#).

### <asid>

The Address Space Identifier of this PLB entry, decimal.

### <vmid>

Virtual Machine Identifier of this PLB entry, decimal. In Tarmac, only printed when `regime` is EL1.

### <tindex>

TIndex used to index the PLB entry, decimal.

### <poindex>

POIndex used to index the PLB entry, decimal.

### <potindex>

POTIndex used to index the PLB entry, decimal.

### <range>

For DPOT accesses, whether the access is to the lower (0) or upper (1) range. Shown as DPOT0 or DPOT1 in output.

## 4.14.13 POE2 table fetch trace

Trace records for IRT fetch, DPOT fetch, POE2 TIndex Transition Table Unprivileged (TTTU) fetch, and POE2 TIndex Transition Table Privileged (TTTP) fetch.

## Output syntax

```
<time> <scale> {<cpu>} IRT<U|P>[TIndex = <tindex>][POIndex = <poindex>]: X = <x>,
  FGDTIndex = <fgdtindex>, POTIndex = <potindex>, nG = <nG>

<time> <scale> {<cpu>} DPOT<range>[POTIndex = <potindex>][POIndex = <poindex>]: R =
  <r>, W = <w>, nG = <nG>

<time> <scale> {<cpu>} TTTU[POTIndex = <potindex>][TIndex = <tindex>]: TCHANGEB =
  <tchangeb>, TCHANGEF = <tchangeb>, nG = <nG>
```

```
<time> <scale> {<cpu>} TTP[POTIndex = <potindex>][TIndex = <tindex>]: TCHANGEB =
<tchangeb>, TCHANGEF = <tchangef>, TENTER = <tenter>, TEXTIT = <textit>, nG = <nG>
```

## Trace field descriptions

### <time>

Either the simulation time timestamp or the instruction count, depending on the value of the TRACE.TarmacTrace.use\_instr\_cnt\_as\_timestamp parameter.

### <scale>

Unit for <time>. ps means simulation time, clk means instruction count.

### <cpu>

Processor that caused the table fetch.

### <U|P>

Whether the access is to the unprivileged (U) or privileged (P) table.

### <tindex>

TIndex used to index the table entry, decimal.

### <poindex>

POIndex used to index the table entry, decimal.

### <x>

Execute permission flag stored in the table entry, boolean.

### <fgdtindex>

FGDTIndex stored in the table entry, decimal.

### <potindex>

POTIndex used to index the table entry, decimal.

### <nG>

Effective value, true if entry is non-global, boolean.

### <range>

For DPOT fetches, whether the access is to the lower (0) or upper (1) range. Shown as DPOT0 or DPOT1 in output.

### <r>

Read permission stored in the table entry, boolean:

0

Read access not permitted

1

Read access permitted

### <w>

Write permission stored in the table entry, boolean:

0

Write access not permitted

**1**

Write access permitted

**<tchangeb>**

TCHANGEB permission flag stored in the table entry, boolean.

**<tchangef>**

TCHANGEF permission flag stored in the table entry, boolean.

**<tenter>**

TENTER permission flag stored in the table entry, boolean.

**<textit>**

TEXTIT permission flag stored in the table entry, boolean.

**Examples**

```
12822 clk IRTU[TIndex = 1][POIndex = 20]: X = 0, FGDTIndex = 0, POTIndex = 8, nG = 0
1100 clk DPOT0[POTIndex = 8][POIndex = 15]: R = 0, W = 0, nG = 0
7610000 ps TTTU[POTIndex = 1][TIndex = 64]: TCHANGEB = 1, TCHANGEF = 0, nG = 0
8700000 ps TTTP[POTIndex = 1][TIndex = 20]: TCHANGEB = 1, TCHANGEF = 1, TENTER = 0,
TEXTIT = 0, nG = 0
```

**4.14.14 Processor memory access trace**

These sources trace processor data accesses.

If FEAT\_MTE or FEAT\_VMTE is implemented:

- MT trace records indicate tag-checking occurs for the following access.
- TR1, TW1, VTR1, and VTW1 trace records indicate a tag load or store.

If FEAT\_HACDBS and FEAT\_HDBSS are implemented:

- MR\_HACDBS or MW\_HACDBS indicate read or write of a HACDBS table.
- MR\_HDBSS or MW\_HDBSS indicate read or write of a HDBSS table.

**Output syntax**

```
<time> <scale> {<cpu>} {GCS} {S1POE2} M<R|W><sz>{<attrib>}{_<atomic_op>} <addr>
<data> {<status>}

<time> <scale> {<cpu>} M<R|W> <HACDBS|HDBSS> <addr> INDEX=<index> IPA=<ipa>
NSIPA=<nsipa> TTWL=<ttwl> VALID=<valid>

<time> <scale> {<cpu>} MT <addr> <tag>

<time> <scale> {<cpu>} {V}T<R|W>1 <addr> {<tag_addr>} <tags> {<status>}
```

## Trace field descriptions

### <time>

Either the simulation time timestamp or the instruction count, depending on the value of the `TRACE.TarmacTrace.use_instr_cnt_as_timestamp` parameter.

### <scale>

Unit for <time>. `ps` means simulation time, `clk` means instruction count.

### <cpu>

Processor, or other component, that gave the instruction.

### <R|W>

#### R

Read access.

#### W

Write access.

### GCS

Indicates a GCS memory access, if `FEAT_GCS` is implemented.

### S1POE2

Indicates a S1POE2 table read.

### <sz>

Size of the data transfer in bytes (1, 2, 4, 8).

### <attrib>

Optional access attribute:

#### X

Exclusive access.

#### T

Translated (unprivileged) access.

#### L

Locked access (SWP, SWPB instructions).

### <atomic\_op>

Atomic operation performed on this memory address:

#### ACC

Accelerator atomics (64-byte atomics. Success is indicated by MU value).

#### ADD

Atomic add operation.

#### BFADD, BFMAX, BFMIN, BFMAXNM, BFMINNM

BFloat16 variants of atomic floating-point memory operations introduced by `FEAT_LSFE`.

**BIC**

Atomic bit clear operation.

**\_CAS<suffix>**

Compare and swap operation, where <suffix> is either *c* or *d*. *\_CASc* shows the value to compare and *\_CASd* shows the value that will be written to memory if the comparison matches.



The value that is stored in memory as a result of a compare and swap operation is shown by an *mx* trace source.

---

**EOR**

Atomic exclusive or operation.

**FADD, FMAX, FMIN, FMAXNM, FMINNM**

AArch64 FP atomics if FEAT\_LSFE is implemented.

**ORR**

Atomic bit set operation.

**RCW{S} {CLR|SWP|SET|CAS<suffix>}**

Read-Check-Write and Read-Check-Write Software instruction variants, as described in the [Arm Architecture Reference Manual for A-profile architecture](#) C3.2.17.6 Read-Check-Write. <suffix> is either *c* or *d*, with the same meaning as in *\_CAS<suffix>*.

**SMAX**

Atomic signed max operation.

**SMIN**

Atomic signed min operation.

**SWP**

Atomic swap operation.

**UMAX**

Atomic unsigned max operation.

**UMIN**

Atomic unsigned min operation.

**<addr>**

Virtual address that is used to access memory. This is the data VA and data PA when virtual tagging is selected. Formatted according to the common address definition, see [TarmacTrace file format](#).



**<data>**

Hexadecimal value of data transferred. The data padding is according to the size of the transfer. Data of 64 bits or more contains an underscore (\_) separator every eight characters (32 bits).

**<status>**

Optional status suffix:

**(ABORTED)**

Access aborted. For example, MTE tag check or memory abort.

**(FAILED)**

Failed exclusive store.

**(UNSUPPORTED)**

Accelerator atomic returned an unsupported result.

**(UNTAGGED RAZ/WI)**

The virtual tag access is **RAZ/WI** due to system register configuration or memory attributes.

**<index>**

Index of the HACDBS or HDBSS entry, decimal.

**<ipa>**

Intermediate physical address translated by the stage 2 descriptor, hexadecimal.

**<nsipa>**

For Secure state, indicates whether the IPA is Secure or Non-secure:

0

Secure IPA.

1

Non-secure IPA.

**<ttl>**

Translation table walk level, decimal.

**<valid>**

Valid entry:

0

Entry is invalid.

1

Entry is valid.

**<tag>**

4-bit Logical Address Tag used for the tag check.

**v**

Indicates that virtual tagging is selected.

**<tag\_addr>**

Tag virtual address of the access. Only appears when virtual tagging is selected. Formatted according to the common address definition, see [TarmacTrace file format](#).

**<tags>**

4-bit Allocation Tag values loaded from or stored to memory.

**Examples**

- 12822 clk MW1 01000000000fff70:0000000fff70 00 (ABORTED)
- 49010000 ps GCS MR8 c00001f0:0000c00001f0\_NS 00000000\_c00001f1
- 12267090000 ps MW8\_RCWSSET 00013000:000000013000 00000000\_00000002
- 12822 clk MT 01000000000fff70:0000000fff70 1
- 1100 clk TR1 000fff70:0000000fff70 0
- 5177 clk TW1 0500000010000000:000010000000 5
- 9056 clk VTR1 0f000000c0bffff0:0000c0bffff0\_NS 4605ffff 0 (UNTAGGED **RAZ/WI**)
- 7254 clk VTW1 0f000000c0000000:0000c0000000\_NS 46000000:000086000000\_NS f
- 7610000 ps MW\_HDBSS 0x000000000001f000\_NS:0x00000000c0000005 INDEX=0  
IPA=0x000000000000c0000 NSIPA=0 TTWL=2 VALID=1
- 8700000 ps MW\_HDBSS ABORTED
- 3340000 ps MR\_HACDBS 0x000000000001f000\_NS:0x00000000c0000005 INDEX=0  
IPA=0x000000000000c0000 NSIPA=0 TTWL=2 VALID=1
- 16100000 ps MR\_HACDBS ABORTED

## 4.14.15 Processor memory update trace

Traces memory update accesses caused by atomic operations.

**Output syntax**

```
<time> <scale> {<cpu>} MU<sz>_<atomic_op> <addr> <data>
```

**Trace field descriptions****<time>**

Either the simulation time timestamp or the instruction count, depending on the value of the `TRACE.TarmacTrace.use_instr_cnt_as_timestamp` parameter.

**<scale>**

Unit for <time>. `ps` means simulation time, `clk` means instruction count.

**<cpu>**

Processor, or other component, that gave the instruction.

**<sz>**

Size of the data transfer in bytes (1, 2, 4, 8, 16).

**<atomic\_op>**

Atomic operation performed on this memory address:

**ADD**

Atomic add operation.

**BFADD, BFMAX, BFMIN, BFMAXNM, BFMINNM**

BFloat16 variants of atomic floating-point memory operations introduced by FEAT\_LSFE.

**BIC**

Atomic bit clear operation.

**CAS**

Atomic compare and swap operation.

**EOR**

Atomic exclusive or operation.

**FADD, FMAX, FMIN, FMAXNM, FMINNM**

AArch64 FP atomics if FEAT\_LSFE is implemented.

**ORR**

Atomic bit set operation.

**SMAX**

Atomic signed max operation.

**SMIN**

Atomic signed min operation.

**SWP**

Atomic swap operation.

**UMAX**

Atomic unsigned max operation.

**UMIN**

Atomic unsigned min operation.

**<addr>**

Physical address that is used to access memory. Formatted according to the common address definition, see [TarmacTrace file format](#).

**<data>**

Hexadecimal value of data that is stored in memory as a result of the atomic operation. Data of 64 bits or more contains an underscore (\_) separator every eight characters (32 bits).

**Examples**

- 12267090000 ps MU8\_CAS 000000013000 00100000\_00000003

- 784 clk MU16\_SWP 00000000b080 dcdcdcdc\_dcdcdcdc\_abababab\_abababab

#### 4.14.16 Program flow trace

If enabled, every executed branch instruction triggers this trace source. It is a more efficient way to reconstruct the program flow than by tracing every instruction.

##### Output syntax

```
<time> <scale> {<cpu>} <FD|FI> (<inst_id>) <addr> <targ_addr> <A|T|X|O>
```

##### Trace field descriptions

###### <time>

Either the simulation time timestamp or the instruction count, depending on the value of the `TRACE.TarmacTrace.use_instr_cnt_as_timestamp` parameter.

###### <scale>

Unit for <time>. `ps` means simulation time, `clk` means instruction count.

###### <cpu>

Processor, or other component, that gave the instruction.

###### <FD|FI>

Program flow change by:

###### **FD**

A direct branch.

###### **FI**

An indirect branch.

###### <inst\_id>

Tick count of this processor, which is equivalent to the number of instructions that are executed, except for certain instructions like `WFI/WFE` (decimal value).

###### <addr>

Fetch source address for this instruction. Formatted according to the common address definition, see [TarmacTrace file format](#).

###### <targ\_addr>

Address (virtual) at which the execution continues. Formatted according to the common address definition.

###### <A|T|X|O>

Instruction set after the branch:

###### **A**

A32.

###### **T**

T32.

**x**  
T32EE.

**o**  
A64

#### 4.14.17 Register trace

Traces all writes to the processor registers.

This trace source includes writes to core registers R0 to R14, X0 to X30, CPSR, and SPSR, VFP registers such as S0 to S31, D0 to D31, FPSCR, and FPEXC, and writes to system registers including CP14, CP15, and GIC. Banked registers are traced separately using the mode as a suffix to the register name, for example r13 (current register R13) and r13\_mon (banked register R13).

#### Output syntax

```
<time> <scale> {<cpu>} R <register> <value>
```

#### Trace field descriptions

##### <time>

Either the simulation time timestamp or the instruction count, depending on the value of the TRACE.TarmacTrace.use\_instr\_cnt\_as\_timestamp parameter.

##### <scale>

Unit for <time>. ps means simulation time, cik means instruction count.

##### <cpu>

Processor, or other component, that gave the instruction.

##### <register>

Register name. Banked core registers can have a mode appended to them with a single underscore. Banked CP14/CP15 registers have \_s or \_ns appended to indicate access of either the Secure or Non-secure banked register.



Note

In Arm®v8 and Arm®v9, when the register name is cpsr:

- In AArch64 state, cpsr is used to trace PSTATE changes. The bit format of <value> follows the SPSR\_ELx AArch64 format.
- In AArch32 state, the bit format of <value> follows the CPSR format.

##### <value>

Hexadecimal value that is written to the register. 64 bits maximum unless FEAT\_D128 is implemented. When tracing 128-bit system register accesses (MSRR/MRRS), print values as two 64-bit halves separated by a colon:

```
<hi64>:<lo64>
```

If the SVE plug-in is loaded in the model, there are additional registers in the program view. The output examples below show how these registers are traced when the value changes. These data values can be very large.

```
8463 clk cpu0 IT (8439) 000282c0:0000152282c0_NS 053fc01f O EL1h_n : SEL
      z31.B,p0,z0.B,z31.B
8463 clk cpu0 R z31 00000000_00000000_00000000_00000000
```

R indicates a register write. z0 to z31 are the vector registers. The written data are hexadecimal digits, which are separated by an underscore every 32 bits. The length of the written data varies with the configuration, depending on the vector length.

```
9756 clk cpu0 IT (9732) 01000074:000011000074_NS 2518e3e0 O EL1t_n : PTRUE p0.B,ALL
9756 clk cpu0 R p0 ffff
```

R indicates a register write. p0 to p15 are the predicate registers. The written data are hexadecimal digits. If they are long enough to require one, the digits are separated by an underscore every 32 bits. The length of the written data varies with the configuration, depending on the vector length. Predicate registers contain 1 bit per byte of vector register length.

#### 4.14.18 SIGNAL trace

Emitted for model signal changes, for example resets.

##### Output syntax

```
<time> <scale> SIGNAL: SIGNAL=<name> STATE=<Y|N>
```

##### Trace field descriptions

###### <time>

Either the simulation time timestamp or the instruction count, depending on the value of the TRACE.TarmacTrace.use\_instr\_cnt\_as\_timestamp parameter.

###### <scale>

Unit for <time>. ps means simulation time, clk means instruction count.

###### <name>

Signal name, for example FIQ, SignalNMIFiq, Reset, Abort, OSUnlockCatch, DebugReset, or CP15SDisable.

###### <Y|N>

Y (asserted) or N (deasserted)

##### Examples

- 0 clk SIGNAL: SIGNAL=DebugReset STATE=N
- 0 clk SIGNAL: SIGNAL=ResetHold STATE=N

### 4.14.19 SPE event trace

Emitted when Statistical Profiling Extension (SPE) packet data is written or discarded, if FEAT\_SPE is implemented.

#### Output syntax

```
<time> <scale> <cpu> SPE <size> <address> <data>
<time> <scale> <cpu> SPE_DISCARDED <size> <address> <data>
```

#### Trace field descriptions

##### <time>

Either the simulation time timestamp or the instruction count, depending on the value of the TRACE.TarmacTrace.use\_instr\_cnt\_as\_timestamp parameter.

##### <scale>

Unit for <time>. ps means simulation time, clk means instruction count.

##### <cpu>

Processor, or other component, that gave the instruction.

##### <size>

The size of <data> in bytes, decimal.

##### <address>

Virtual address of the start of the buffer write, hexadecimal.

##### <data>

SPE packet data written or discarded, hexadecimal. Every 8 bytes are separated by an underscore.

#### Examples

- 8057 clk cpu0 SPE 59 000023000076  
0000000020013643\_71000099000198a0\_00000001028a34b0\_9abcdef065123456\_7864164200009a80\_000000604
- 84475 clk cpu0 SPE\_DISCARDED 56 000080000000  
0000000000000000\_7100009900019840\_00000001003b58b0\_9abcdef065164200\_4300009a00000000\_22869d5cb

### 4.14.20 TLB trace

Traces TLB entries that are filled and evicted by the processor.

#### Output syntax

```
<time> <scale> <cpu> <TLB|WALKCACHE> FILL <id> <size> <virtualregime>:<paddr>
{<memtype>} {<attr>=<value>}+
<time> <scale> <cpu> <TLB|WALKCACHE> EVICT <id> <size> <virtualregime>
```

## Trace field descriptions

### <time>

Either the simulation time timestamp or the instruction count, depending on the value of the `TRACE.TarmacTrace.use_instr_cnt_as_timestamp` parameter.

### <scale>

Unit for <time>. `ps` means simulation time, `clk` means instruction count.

### <cpu>

Processor, or other component, that gave the instruction.

### <id>

Identifies which TLB or walk cache to trace.

### <size>

Size of the region being mapped.

### <virtualregime>

Virtual address and regime of the region being mapped, formatted according to the common virtual regime definition, see [TarmacTrace file format](#).

### <paddr>

Physical base address of mapped region, formatted according to the common address definition, see [TarmacTrace file format](#).

### <memtype>

For TLB entries, the memory type of the result. One of the following options:

**Device- [G|nG] [R|nR] [E|nE] {(<alias>) }**

Device memory, where:

**[G|nG]**

Gathering or nongathering.

**[R|nR]**

Reordering or nonreordering.

**[E|nE]**

Early write acknowledgement or not.

**<alias>**

Device-nGnRnE was previously known as StronglyOrdered.

**Normal [NonShareable|Shareable] Inner=<cachetype> Outer=<cachetype>**

Normal memory, where:

**[NonShareable|Shareable]**

Shareability

**<cachetype>**

[NonCacheable|WriteBack|WriteThrough] {NonReadAllocate} {Non}  
{WriteAllocate}



**[NonCacheable|WriteBack|WriteThrough]**

Cacheability

**{NonReadAllocate}**

For cacheable memory, Read allocate hint. (Read allocate is assumed if not specified.)

**{Non}{WriteAllocate}**

For cacheable memory, Write allocate hint.

## 4.14.21 Translation table walk trace

Traces all translation table walks initiated by the processor.

### Output syntax

```
<time> <scale> <cpu> {<SPE|TRBE|S1POE2|VMTE>} <TTW|TTU|TTU_HACDBS> <D|I>TLB <VMSA|
LPAE|D128> <stage>:<level> <address> <data> : <result>
```

### Trace field descriptions

#### <time>

Either the simulation time timestamp or the instruction count, depending on the value of the `TRACE.TarmacTrace.use_instr_cnt_as_timestamp` parameter.

#### <scale>

Unit for <time>. `ps` means simulation time, `clk` means instruction count.

#### <cpu>

Processor, or other component, that gave the instruction.

#### <SPE|TRBE|S1POE2|VMTE>

If present, indicates which feature triggered the table walk. One of:

##### **SPE**

Statistical Profiling Extension requestor

##### **TRBE**

Trace Buffer Extension requestor

##### **S1POE2**

S1POE2 table walk requestor

##### **VMTE**

VMTE requestor

If not present, the table walk happened because of the default reason, which can be one of the following:

- Instruction fetch
- Explicit memory access by a load/store instruction
- Explicit cache maintenance operation

- Explicit prefetch.

**<TTW|TTU|TTU\_HACDBS>**

Translation table walks (reads), translation table updates (writes), or translation table updates related to FEAT\_HACDBS.

**<D|I>**

Data (D) or instruction (I) side TLB.

**<VMSA|LPAE|D128>**

Format of the translation table entries:

**VMSA**

VMSAv8-32 short-descriptor translation table format

**LPAE**

Either:

- VMSAv8-32 long-descriptor translation table format in AArch32
- VMSAv8-64 descriptor format in AArch64

**D128**

VMSAv9-128 table descriptor format if FEAT\_LVA is implemented

**<stage>**

Walk stage, within the range 1-2.

**<level>**

Walk level, within the range 1-3.

**<address>**

Physical address of lookup in hexadecimal.

**<data>**

Raw translation table entry in hexadecimal.

**<result>**

Parsed result. One of the following options:

**ABORTED**

The memory access caused a synchronous abort and no data was returned.

**FAULT**

The data that was returned is not valid for this stage and level.

**RESERVED**

The data that was returned is not valid for this stage and level.

**TABLE {<attr>=<value>}+**

Pointer to the next level of lookup, in LPAE format.

**BLOCK {<attr>=<value>}**

LPAE region descriptor.

**SUPERSECTION {<attr>=<value>}**

VMSA region descriptor.

**SECTION {<attr>=<value>}**

VMSA region descriptor.

**PAGETABLE {<attr>=<value>}**

Pointer to the next level of lookup, in VMSA format.

**LARGEPAGE {<attr>=<value>}**

VMSA region descriptor.

**SMALLPAGE {<attr>=<value>}**

VMSA region descriptor.

## 4.14.22 vMTE tag VA access trace

Virtual tag access trace. It is emitted by VMTE tag operations.

### Output syntax

```
<time> <scale> {<cpu>} VMT <vaddr>:<tag_paddr>_<tag_pas> <vaddr_vtag> <(hi) | (lo)>
<vtag>
```

### Trace field descriptions

#### <time>

Either the simulation time timestamp or the instruction count, depending on the value of the `TRACE.TarmacTrace.use_instr_cnt_as_timestamp` parameter.

#### <scale>

Unit for <time>. `ps` means simulation time, `clk` means instruction count.

#### <cpu>

Processor that gave the instruction.

#### <vaddr>

Data virtual address, hexadecimal.

#### <tag\_paddr>

Physical address of the virtual tag, hexadecimal.

#### <tag\_pas>

PAS of the virtual tag.

#### <vaddr\_vtag>

Virtual address of the virtual tag, hexadecimal.

#### <(hi) | (lo)>

Which half of the tag data applies. This value is derived from bit [4] of `vaddr` in the Tarmac formatter.

**vtag**

Virtual tag value, hexadecimal.

**Example**

```
234 clk VMT 00038100:000000038100_NS 7849AFB89998080990DD8 (hi) a
```

## 4.15 ToggleMTIPlugin

ToggleMTIPlugin is an MTI plug-in that can be used to limit the generation of trace by another plug-in to specific areas of interest.

Generating trace output throughout a simulation session can reduce simulation speed and result in very large trace files. ToggleMTIPlugin helps to avoid these problems by enabling you to toggle trace generation during the simulation. Toggling trace means that if trace is on, it is turned off, and vice versa.

ToggleMTIPlugin can be used with the following plug-ins:

- ASTFplugin
- GenericTrace
- TarmacTrace

### 4.15.1 ToggleMTIPlugin - parameters

Each parameter is prefixed with `TRACE.ToggleMTIPlugin`, for example:

```
TRACE.ToggleMTIPlugin.diagnostics
```

Name	Type	Default value	Allowed values	When set	Description
diagnostics	Bool	0x0	true, false	Init time	Print diagnostics
disable_mti_from_start	Bool	0x0	true, false	Init time	Enable or disable MTI callbacks from start of simulation
disable_mti_runtime	Bool	0x0	true, false	Runtime	Enable or disable MTI callbacks at runtime
hlt_imm16	Int	0xf000	0x0 - 0xffff	Init time	16-bit integer used in HLT instruction meant to be used by this plugin
use_hlt	Bool	0x1	true, false	Init time	If true, use HLT #imm16 instruction to toggle MTI behavior

## 4.15.2 How to use ToggleMTIPlugin

As with other plug-ins, load ToggleMTIPlugin using the `--plugin` command-line option when launching the model.



Note

- When loading ToggleMTIPlugin and any other trace plug-ins using the `--plugin` option, ToggleMTIPlugin must be the last plug-in to be specified on the command line.
- We recommend you disable trace generation from the start of the simulation, using the plug-in parameter `disable_mti_from_start=1`, then enable it when execution reaches the region of interest.

There are two alternative ways to use ToggleMTIPlugin. You cannot use both in the same simulation session. Use the `use_hlt` plug-in parameter to control which one to use:

- `use_hlt = 1`

To use this method, set the `hlt_imm16` plug-in parameter to an integer value. The application will use this value as the operand in `HLT` instructions to toggle MTI callbacks.

You must also set the following parameters on the core model that is running the application:

### **enable\_trace\_special\_hlt\_imm16**

Set to true to enable the parameter `trace_special_hlt_imm16`.

### **trace\_special\_hlt\_imm16**

Specifies the integer value that is used as the operand to `HLT` instructions to cause the usual `HLT` execution to be skipped. If the value matches the value specified in the `hlt_imm16` plug-in parameter, tracing is toggled.

- `use_hlt = 0`

To use this method, set the runtime plug-in parameter `disable_mti_runtime` during the simulation session to either true to disable tracing, or false to enable tracing. Changes to the `disable_mti_runtime` parameter are ignored unless `use_hlt` is zero.

To change `disable_mti_runtime` at runtime, use a debugger, for example Iris Monitor or use the `iris.debug` Python module. The example Python script `$PVLIB_HOME/examples/pyIris/inst_count_trace_control.py`, demonstrates how to do this.

## 5. SystemC Export

This section describes the SystemC eXport (SCX) API provided by Fast Models Exported Virtual Subsystems (EVSs). Each description of a class or function includes the C++ declaration and the use constraints.

### 5.1 About SystemC Export

SystemC Export wraps the components of a SystemC-based virtual platform into an Exported Virtual Subsystem (EVS). Multiple Instantiation (MI) enables the generation and integration of multiple EVS instances into a single SystemC simulation.

SystemC Export enables the generation of EVSs as first-class SystemC components:

- Capable of running any number of instances, alongside other EVSs.
- Providing one `sc_THREAD` per core component (that is, one `sc_THREAD` per core component in a cluster Code Translation (CT) model).

SystemC Export enables the generation and integration of multiple EVS instances into a virtual platform with SystemC as the single simulation domain. A single EVS can appear in multiple virtual platforms. Equally, multiple EVSs can combine to create a single platform. A platform that consists of multiple EVSs is called an SVP (SystemC Virtual Platform).

SystemC components (including Fast Models ones) can exchange data via the Direct Memory Interface (DMI) or normal (blocking) Transaction Level Modeling (TLM) transactions.

Fast Models supports SystemC 2.3.4, including integrated TLM 2.0.6. In this version, the TLM and SystemC headers are in the same place, and some filenames are different.

Before using SimGen to build a SystemC simulation, the environment variable `SYSTEMC_HOME` must be set to the directory containing the Accellera SystemC library installation.

#### Related information

[Fast Models Reference Guide](#)

[Accellera Systems Initiative \(ASI\)](#)

[IEEE Std 1666-2005, SystemC Language Reference Manual, 31 March 2006](#)

[Accellera, TLM 2.0 Language Reference Manual, July 2009](#)

### 5.2 Bridging between LISA+ and SystemC

Bridges are components that convert transactions from one protocol to another.

In Fast Models, Bridge components allow conversion between [LISA+ protocols](#) and their SystemC equivalents. For example between:

- PVBUS and AMBAPV.
- Signal and AMBAPVSignal.
- StateSignal and AMBAPVSignalState.
- Value(\_64) and AMBAPVValue(64).
- ValueState(\_64) and AMBAPVValueState(64).

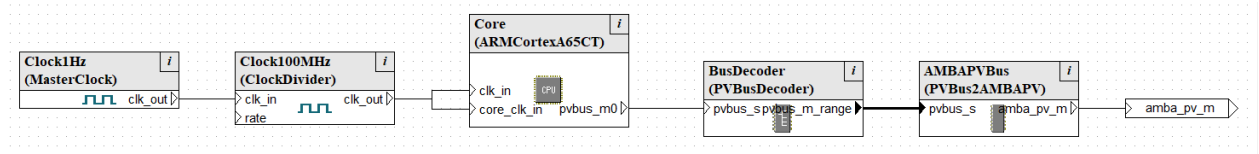
Bridge components have the LISA+ property `component_type = "Bridge"`. LISA+ source for them is located in `$PVLIB_HOME/examples/SystemCExport/Bridges/`.

The examples under `$PVLIB_HOME/examples/SystemCExport/EVS_Platforms/EVS_Dhrystone/` show how bridging works to integrate LISA+ models with AMBAPV.

They consist of an Arm core and some simple peripherals that are written in LISA+, some memory that is defined in SystemC, and a bridge between the exported LISA+ subsystem and the SystemC code.

The following System Canvas block diagram shows the Fast Models LISA+ components that are defined in the top-level Dhrystone component and the connections between them. The `amba_pv_m` port at the right-hand side is used to connect the Dhrystone component to the memory, which is defined in SystemC:

**Figure 5-1: System Canvas block diagram for EVS\_Dhrystone**



The [PVBUS2AMBAPV](#) component is a bridge that converts signals from the PVBUS protocol to the AMBA-PV protocol. After exporting the Dhrystone component as an EVS, the `amba_pv_m` port can be connected to a SystemC component, in this example, to the slave port of the memory model, in `main.cpp`, as follows:

```
amba_pv::amba_pv_memory<64> memory("Memory", 0x34000100);
scx_evs_dhrystone dhrystone("Dhrystone");
...
dhrystone.amba_pv_m(memory.amba_pv_s);
```

The AMBAPV protocols and components are designed to interface with the AMBA PV library for TLM 2.0. Fast Models provides this library as a standard way of mapping the AMBA buses on top of SystemC TLM 2.0.

For more information about AMBA PV, see:

- [AMBA-PV Extensions to TLM 2.0 Reference Guide](#)
- [AMBA-PV Extensions to TLM User Guide](#)

For more information about SystemC TLM 2.0, see <https://www.accellera.org>.

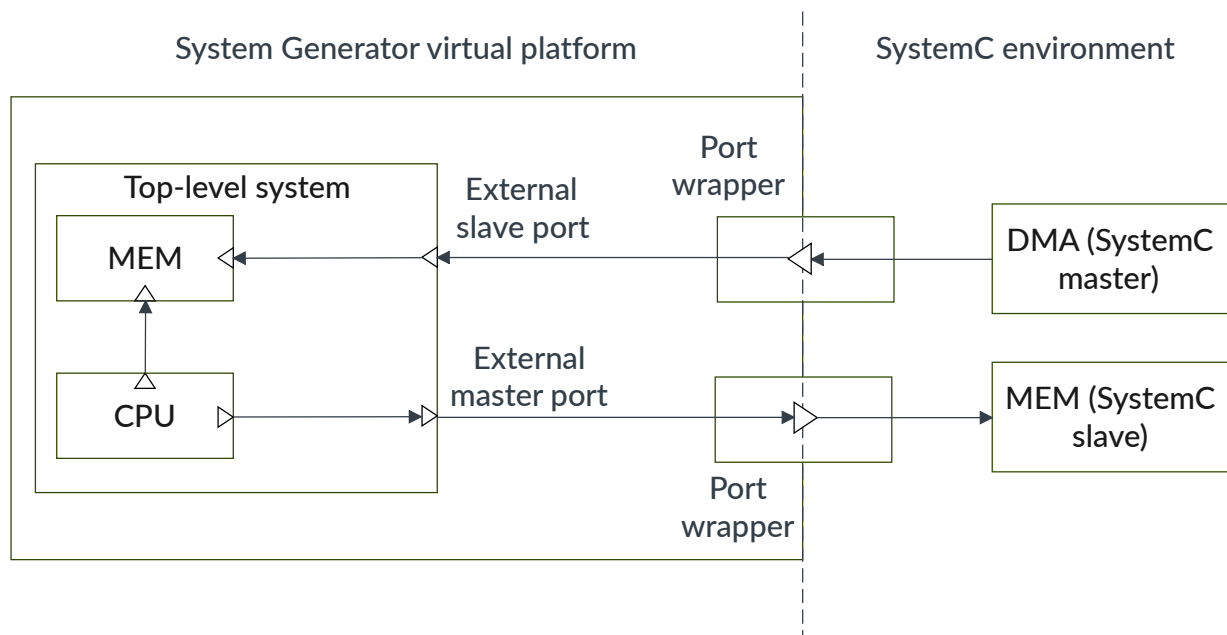
## 5.3 SystemC Export generated ports

The generated SystemC component must have SystemC ports to communicate with the SystemC world. The SystemC export feature automatically generates port wrappers that bind the Fast Models virtual platform to the SystemC domain.



Although it is possible to export your own protocols, Arm strongly recommends using the AMBA-PV protocols provided and bridge from these in SystemC, if needed.

**Figure 5-2: Port wrappers connect Fast Models and SystemC components**



Each master port and slave port in the Fast Models top-level component result in a master port and slave port (export) on the SystemC side.

For Fast Models to instantiate and use the ports, it requires protocol definitions that:

- Correspond to the equivalent SystemC port classes.
- Refer to the names of these SystemC port classes.

This effectively describes the mapping from Fast Models port types (protocols) to SystemC port types (port classes).



## 5.4 SystemC Export limitations

This section describes the limitations of SystemC Export.

- Reentrancy occurs when a component in an EVS issues a blocking transaction to a SystemC peripheral that in turn generates another blocking transaction back into the same component. This generation might come directly or indirectly from a call to `wait()` or by another SystemC peripheral.

Virtual platforms including EVSs that comprise a processor model do support such reentrancy.

For models that do not support reentrancy, the virtual platform might show unpredictable behavior because of racing within the EVS component.

- Fast Models only supports calling `wait()` on bus transactions.

When a SystemC peripheral must really issue a `wait()` in reaction to a signal that is changing, buffer the signal in the bridge between the EVS and SystemC. On the next activation of the bridge, set the signal with the thread context of the EVS.



The EVS runs in a temporally-decoupled mode using a time quantum. Transaction Level Modeling (TLM) 2.0 targets using the Loosely-Timed coding style do not call `wait()`.

- 
- EVS core components use code translation for speed. Not enabling Direct Memory Interface (DMI) reduces performance.

The core components in EVSs use code translation for high simulation speed. They fetch data from external memory to translate it into host machine code. Changing the memory contents outside of the scope of the core makes the data inconsistent.

Enable DMI accesses to instruction memory to avoid dramatic performance reductions. Otherwise, EVSs:

- Model all accesses.
  - Perform multiple spurious transactions.
  - Translate code per instruction not per block of instructions.
- Multiple EVSs in a virtual platform must have been made with the same version of Fast Models. Integrating EVSs from different versions of Fast Models might result in unpredictable behavior.

## 5.5 scx API

The main header file for SystemC export support is `$PVLIB_HOME/include/fmruntime/scx/scx.h`.

### **scx\_enable\_iris\_server()**

Defined in `scx_iris.h`.

The `scx_enable_iris_server()` function is overloaded. These are the different versions of this function:

- ```
SCX_API void scx::scx_enable_iris_server(const std::string &connection_spec)
```

Starts an Iris server.

An empty string stops the server. Specifying `"help"` prints the syntax and semantics of all supported server types.

For example:

- TCP server: `"tcpserver,port=7100,endport=7355,allowRemote"`
- UNIX domain socket connection: `"socketfd=42"`
- Stop Iris server: `""`

Parameters:

#### **connection\_spec**

String specifying the connection specifications.

- ```
SCX_API void scx::scx_enable_iris_server(bool enable=true)
```

Specifies whether to start an Iris TCP server.

If a server is started, it listens on a port in the range `[DefaultIrisServerPortMin, DefaultIrisServerPortMax]`

Parameters:

#### **enable**

Set to true to enable the Iris server or to false to stop the Iris server.

### **scx\_set\_iris\_server\_port\_range()**

Defined in `scx_iris.h`.

Sets the range of ports to scan when starting an Iris server.

The first available port found is used.

```
SCX_API void scx::scx_set_iris_server_port_range(uint16_t port_min, uint16_t port_max)
```

Parameters:

**port\_max**

Specifies the upper port number.

**port\_min**

Specifies the lower port number.



Note

This function only takes effect if you call it before starting the Iris server.

---

### **scx\_get\_iris\_server\_port()**

Defined in `scx_iris.h`.

Returns the Iris TCP port number assigned after the Iris server has started. Otherwise returns zero.

```
SCX_API uint16_t scx::scx_get_iris_server_port()
```

### **scx\_set\_iris\_server\_port()**

Defined in `scx_iris.h`.

Sets a specific port to use for the Iris server.

```
void scx::scx_set_iris_server_port(uint16_t port)
```

Parameters:

**port**

Specifies the only port to try to use. If it is unavailable, the server does not start.

---



Note

This function only takes effect if you call it before starting the Iris server.

---

### **scx\_enable\_iris\_log()**

Defined in `scx_iris.h`.

Specifies the Iris log level.

The possible values are:

- 0: Logging is disabled. This is the default value.
- 1: Log messages use a compact, single-line format.

- 2: Log messages use a single-line, pseudo-JSON format.
- 3: Log messages use a more readable multi-line, pseudo-JSON format.
- 4: As 3 but also prints the U64JSON hex value of the message.

```
SCX_API void scx::scx_enable_iris_log(unsigned level=0)
```

Parameters:

**level**

Specifies the log level.

### **scx\_get\_iris\_connection\_interface()**

Defined in `scx_iris.h`.

Retrieves the `IrisConnectionInterface` for the simulation. This can be used to create and register `IrisInstances`.

```
SCX_API iris::IrisConnectionInterface * scx::scx_get_iris_connection_interface()
```

### **scx\_sync()**

Defined in `scx_sched.h`.

Adds a future synchronization point in time.

SystemC components call this function to provide a hint to the scheduler implementation when a potentially useful system synchronization point will occur in the future. The scheduler uses this information to determine the quantum size of threads as they are scheduled.

Threads that already ran their quantum are unaffected. All other threads, including the current thread, will run until the synchronization point specified by `sync_time`.

Calling `scx_sync()` again adds another synchronization point.

Synchronization points are automatically removed when the simulation time reaches them.

```
SCX_API void scx::scx_sync(double sync_time)
```

Parameters:

**sync\_time**

Specifies the time of the future synchronization point relative to the start of the current quantum, measured in seconds.



Arm deprecates this function. Use IEEE 1666 SystemC 2011 `sc_core::sc_prim_channel:async_request_update()` instead.

## **scx\_set\_min\_sync\_latency()**

Defined in `scx_sched.h`.

The `scx_set_min_sync_latency()` function is overloaded. These are the different versions of this function:

- `SCX_API void scx::scx_set_min_sync_latency(double t)`

Sets the minimum synchronization latency for this scheduler.

The minimum synchronization latency helps to ensure that sufficient simulated time has passed between two synchronization points for synchronization to be efficient.

A small latency increases accuracy but decreases simulation speed. A large latency decreases accuracy but increases simulation speed.

The scheduler uses this information to compute the next synchronization point as returned by `sg::SchedulerInterfaceForComponents::getNextSyncPoint()`.

Parameters:

**t**

Specifies the minimum synchronization latency, measured in seconds.

- `SCX_API void scx::scx_set_min_sync_latency(sg::ticks_t t)`

Sets the minimum synchronization latency for this scheduler.

The minimum synchronization latency helps to ensure that sufficient simulated time has passed between two synchronization points for synchronization to be efficient.

A small latency increases accuracy but decreases simulation speed. A large latency decreases accuracy but increases simulation speed.

The scheduler uses this information to compute the next synchronization point as returned by `sg::SchedulerInterfaceForComponents::getNextSyncPoint()`.

Parameters:

**t**

Specifies the minimum synchronization latency, measured in ticks.

## **scx\_get\_min\_sync\_latency()**

Defined in `scx_sched.h`.

The `scx_get_min_sync_latency()` function is overloaded. These are the different versions of this function:

- `SCX_API double scx::scx_get_min_sync_latency()`

Returns the minimum synchronization latency for this scheduler, measured in seconds.

- `SCX_API sg::ticks_t scx::scx_get_min_sync_latency(sg::Tag< sg::ticks_t > *)`

Returns the minimum synchronization latency for this scheduler, measured in ticks.

### **scx\_simlimit()**

Defined in `scx_sched.h`.

Sets the maximum number of seconds to simulate.

```
SCX_API void scx::scx_simlimit(double t)
```

Parameters:

**t**

Number of seconds to simulate. Defaults to unlimited.

### **scx\_cyclelimit()**

Defined in `scx_sched.h`.

Sets the maximum number of cycles to run.

```
SCX_API void scx::scx_cyclelimit(uint64_t cycles)
```

Parameters:

**cycles**

Number of cycles to run. Defaults to unlimited.

### **scx\_create\_default\_scheduler\_mapping()**

Defined in `scx_sched.h`.

Returns a pointer to a new instance of the default implementation of the scheduler mapping provided with Fast Models.

```
sg::SchedulerInterfaceForComponents * scx::  
scx_create_default_scheduler_mapping(scx_simcontrol_if *simcontrol)
```

Parameters:

**simcontrol**

Pointer to an existing simulation controller. If this is `NULL`, this function returns `NULL`.

**scx\_get\_curr\_scheduler\_mapping()**

Defined in `scx_sched.h`.

Returns a pointer to the current implementation of the scheduler-mapping interface.

```
SCX_API sg::SchedulerInterfaceForComponents * scx::  
scx_get_curr_scheduler_mapping()
```

**scx\_initialize()**

Defined in `scx_simconfig.h`.

Initializes the simulation.

Use this function to initialize the simulation. The simulation must be initialized before constructing any exported subsystem.

```
SCX_API void scx::scx_initialize(const std::string &id, scx_simcontrol_if  
*ctrl=scx_get_default_simcontrol())
```

Parameters:

**ctrl**

Specifies a pointer to the simulation control implementation. Defaults to the one provided by Fast Models.

**id**

Specifies an identifier for this simulation.



It is recommended to specify an identifier that is unique across all simulations running on the same host.

---

**scx\_set\_single\_evs()**

Defined in `scx_simconfig.h`.

Sets the simulation engine to accept a single EVS only. The EVS name prefix is removed from all parameter names. Call this function immediately after `scx_initialize()`.

```
SCX_API void scx::scx_set_single_evs()
```

**scx\_load\_application()**

Defined in `scx_simconfig.h`.

Loads the given application in the memory of the given instance.

The parameter `instance` must start with an EVS instance name, or with `'*` to load the `application` into the instance on all EVSs in the platform.

To load the same application on all cores of an SMP processor, specify `"*` for the core instead of its index, in parameter `instance`.

The earliest the application can be loaded is in the `start_of_simulation()` callback.

```
SCX_API void scx::scx_load_application(const std::string &instance, const
std::string &application)
```

Parameters:

**application**

Specifies the application to load.

**instance**

Specifies the name of the instance to load the application into.

**scx\_load\_application\_all()**

Defined in `scx_simconfig.h`.

Loads the given `application` in the memory of all instances that execute software, across all EVSs in the platform.

The earliest the application can be loaded is in the `start_of_simulation()` callback.

```
SCX_API void scx::scx_load_application_all(const std::string &application)
```

Parameters:

**application**

Specifies the application to load.

**scx\_load\_data()**

Defined in `scx_simconfig.h`.

Loads the given raw data on the given `instance` at the given memory address.

The parameter `instance` must start with an EVS instance name, or with `'*` to load the raw data into the given instance on all EVSs in the platform.

On an SMP processor, if the parameter `instance` specifies `'*` for the core instead of its index, the given raw data is only loaded on the first core.

The earliest the raw data can be loaded is in the `start_of_simulation()` callback.

```
SCX_API void scx::scx_load_data(const std::string &instance, const std::string
&data, const std::string &address)
```



Parameters:

**address**

Specifies the memory address at which to load the given raw data. This parameter might start with a memory space specifier.

**data**

Specifies the filename of the raw data to load.

**instance**

Specifies the name of the instance to load into.

### **scx\_load\_data\_all()**

Defined in `scx_simconfig.h`.

Loads the given raw data on all instances that execute software, across all EVSs in the platform, at the given memory address.

On an SMP processor, the given raw data is only loaded on the first core.

The earliest the raw data can be loaded is in the `start_of_simulation()` callback.

```
SCX_API void scx::scx_load_data_all(const std::string &data, const std::string
&address)
```

Parameters:

**address**

Specifies the memory address at which to load the given raw data. This parameter might start with a memory space specifier.

**data**

Specifies the filename of the raw data to load.

### **scx\_set\_parameter()**

Defined in `scx_simconfig.h`.

The `scx_set_parameter()` function is overloaded. These are the different versions of this function:

- ```
SCX_API bool scx::scx_set_parameter(const std::string &name, const std::string
&value)
```

Sets the value of the given parameter.

Use this function to set a parameter for a component that is present in the EVS or for a plug-in.

The parameter `name` must start with an EVS instance name to set the parameter for a specific EVS, or with "\*" to set the parameter for all EVSs in the platform. To set a plug-in parameter, `name` must start with a plug-in prefix, which defaults to "TRACE".

Parameters:

**name**

Specifies the name of the parameter.

**value**

Specifies the value of the parameter.

Returns `true` if the parameter exists, `false` otherwise.



Note

Changes made to parameters within System Canvas take precedence over changes made with this function.



Note

You can use this function to set parameters during the construction phase and before the elaboration phase. Calls to this function after the construction phase are ignored. You can change runtime parameters after the construction phase using the debug interface.



Note

Plug-ins must be loaded before calling any of the platform parameter functions, otherwise their parameters will not be available. Any plug-in specified after the first call to any platform parameter function is ignored.

- ```
bool scx::scx_set_parameter(const std::string &name, T value)
```

Sets the value of the given parameter.

Use this function to set a parameter for a component that is present in the EVS or for a plug-in.

The parameter `name` must start with an EVS instance name to set the parameter for a specific EVS, or with `"*"` to set the parameter for all EVSs in the platform. To set a plug-in parameter, `name` must start with a plug-in prefix, which defaults to `"TRACE"`.

Parameters:

**name**

Specifies the name of the parameter.

**value**

Specifies the value of the parameter.

Returns `true` if the parameter exists, `false` otherwise.



Note

Changes made to parameters within System Canvas take precedence over changes made with this function.

**Note**

You can use this function to set parameters during the construction phase and before the elaboration phase. Calls to this function after the construction phase are ignored. You can change runtime parameters after the construction phase using the debug interface.

**Note**

Plug-ins must be loaded before calling any of the platform parameter functions, otherwise their parameters will not be available. Any plug-in specified after the first call to any platform parameter function is ignored.

- ```
bool scx::scx_set_parameter(const std::string &name, char value)
```

Sets the value of the given 8-bit parameter.

Use this function to set a parameter for a component that is present in the EVS or for a plug-in.

The parameter `name` must start with an EVS instance name to set the parameter for a specific EVS, or with "\*" to set the parameter for all EVSs in the platform. To set a plug-in parameter, it must start with a plug-in prefix, which defaults to "TRACE".

Parameters:

**name**

Specifies the name of the parameter.

**value**

Specifies the 8-bit value of the parameter.

Returns `true` if the parameter exists, `false` otherwise.

**Note**

Changes made to parameters within System Canvas take precedence over changes made with this function.

**Note**

You can use this function to set parameters during the construction phase and before the elaboration phase. Calls to this function after the construction phase are ignored. You can change runtime parameters after the construction phase using the debug interface.

**Note**

Plug-ins must be loaded before calling any of the platform parameter functions, otherwise their parameters will not be available. Any plug-in specified after the first call to any platform parameter function is ignored.

- ```
bool scx::scx_set_parameter(const std::string &name, signed char value)
```

Sets the value of a given 8-bit signed parameter.

Use this function to set a parameter for a component that is present in the EVS or for a plug-in.

The parameter `name` must start with an EVS instance name to set the parameter for a specific EVS, or with "\*" to set the parameter for all EVSs in the platform. To set a plug-in parameter, it must start with a plug-in prefix, which defaults to "TRACE".

Parameters:

**name**

Specifies the name of the parameter.

**value**

Specifies the 8-bit signed value of the parameter.

Returns `true` if the parameter exists, `false` otherwise.

**Note**

Changes made to parameters within System Canvas take precedence over changes made with this function.

**Note**

You can use this function to set parameters during the construction phase and before the elaboration phase. Calls to this function after the construction phase are ignored. You can change runtime parameters after the construction phase using the debug interface.

**Note**

Plug-ins must be loaded before calling any of the platform parameter functions, otherwise their parameters will not be available. Any plug-in specified after the first call to any platform parameter function is ignored.

- ```
bool scx::scx_set_parameter(const std::string &name, unsigned char value)
```

Sets the value of a given 8-bit unsigned parameter.

Use this function to set a parameter for a component that is present in the EVS or for a plug-in.

The parameter `name` must start with an EVS instance name to set the parameter for a specific EVS, or with "\*" to set the parameter for all EVSs in the platform. To set a plug-in parameter, it must start with a plug-in prefix, which defaults to "TRACE".

Parameters:

**name**

Specifies the name of the parameter.

**value**

Specifies the 8-bit unsigned value of the parameter.

Returns `true` if the parameter exists, `false` otherwise.



Changes made to parameters within System Canvas take precedence over changes made with this function.



You can use this function to set parameters during the construction phase and before the elaboration phase. Calls to this function after the construction phase are ignored. You can change runtime parameters after the construction phase using the debug interface.



Plug-ins must be loaded before calling any of the platform parameter functions, otherwise their parameters will not be available. Any plug-in specified after the first call to any platform parameter function is ignored.

## **scx\_get\_parameter()**

Defined in `scx_simconfig.h`.

The `scx_get_parameter()` function is overloaded. These are the different versions of this function:

- ```
SCX_API bool scx::scx_get_parameter(const std::string &name, std::string &value)
```

Retrieves the value of a given parameter.

Use this function to get a parameter for a component that is present in the EVS or for a plug-in.

The parameter `name` must start with an EVS instance name to retrieve an EVS parameter or with a plug-in prefix, which defaults to "TRACE", to retrieve a plug-in parameter.

Parameters:

**name**

Specifies the name of the parameter.

**value**

Specifies a reference to the value of the parameter.

Returns `true` if the parameter exists, `false` otherwise.



Plug-ins must be loaded before calling any of the platform parameter functions, otherwise their parameters will not be available. Any plug-in specified after the first call to any platform parameter function is ignored.

- ```
bool scx::scx_get_parameter(const std::string &name, T &value)
```

Retrieves the value of a given parameter.

Use this function to get a parameter for a component that is present in the EVS or for a plug-in.

The parameter `name` must start with an EVS instance name to retrieve an EVS parameter or with a plug-in prefix, which defaults to `"TRACE"`, to retrieve a plug-in parameter.

Parameters:

**name**

Specifies the name of the parameter.

**value**

Specifies a reference to the value of the parameter.

Returns `true` if the parameter exists, `false` otherwise.



Plug-ins must be loaded before calling any of the platform parameter functions, otherwise their parameters will not be available. Any plug-in specified after the first call to any platform parameter function is ignored.

- ```
SCX_API bool scx::scx_get_parameter(const std::string &, bool &)
```

Retrieves the value of a given parameter as a `bool`.

- ```
SCX_API bool scx::scx_get_parameter(const std::string &, short &)
```

Retrieves the value of a given parameter as a `short`.

- ```
SCX_API bool scx::scx_get_parameter(const std::string &, unsigned short &)
```

Retrieves the value of a given parameter as an unsigned short.

- ```
SCX_API bool scx::scx_get_parameter(const std::string &, int &)
```

Retrieves the value of a given parameter as an int.

- ```
SCX_API bool scx::scx_get_parameter(const std::string &, unsigned int &)
```

Retrieves the value of a given parameter as an unsigned int.

- ```
SCX_API bool scx::scx_get_parameter(const std::string &, long &)
```

Retrieves the value of a given parameter as a long.

- ```
SCX_API bool scx::scx_get_parameter(const std::string &, unsigned long &)
```

Retrieves the value of a given parameter as an unsigned long.

- ```
SCX_API bool scx::scx_get_parameter(const std::string &, long long &)
```

Retrieves the value of a given parameter as a long long.

- ```
SCX_API bool scx::scx_get_parameter(const std::string &, unsigned long long &)
```

Retrieves the value of a given parameter as an unsigned long long.

- ```
std::string scx::scx_get_parameter(const std::string &name)
```

Retrieves the value of a given parameter.

Use this function to get a parameter for a component that is present in the EVS or for a plug-in.

The parameter `name` must start with an EVS instance name to retrieve an EVS parameter or with a plug-in prefix, which defaults to "TRACE" to retrieve a plug-in parameter.

Parameters:

**name**

Specifies the name of the parameter to be retrieved.

Returns the value of the parameter if it exists, empty string ( "") otherwise.



Plug-ins must be loaded before calling any of the platform parameter functions, otherwise their parameters will not be available. Any plug-in specified after the first call to any platform parameter function is ignored.

---

## **scx\_get\_parameter\_list()**

Defined in `scx_simconfig.h`.

Retrieves a list of all parameters for all components present in all EVSs and for all plug-ins in this simulation.

EVS parameter names start with an EVS instance name and plug-in parameter names start with a plug-in prefix, which defaults to "TRACE".

```
SCX_API std::map< std::string, std::string > scx::scx_get_parameter_list()
```



Note

Plug-ins must be loaded before calling any of the platform parameter functions, otherwise their parameters will not be available. Any plug-in specified after the first call to any platform parameter function is ignored.



Note

If `scx_set_parameter()` is called after the simulation elaboration phase, `scx_get_parameter_list()` returns the new value, although it is not set in the model.

## **scx\_get\_parameter\_infos()**

Defined in `scx_simconfig.h`.

Retrieves a list of all parameter descriptions for all components present in all EVSs and for all plug-ins in this simulation.

EVS parameter names start with an EVS instance name and plug-in parameter names start with a plug-in prefix, which defaults to "TRACE".

```
SCX_API std::map< std::string, std::string > scx::scx_get_parameter_infos()
```



Note

Plug-ins must be loaded before calling any of the platform parameter functions, otherwise their parameters will not be available. Any plug-in specified after the first call to any platform parameter function is ignored.

## **scx\_get\_cadi\_parameter\_infos()**

Defined in `scx_simconfig.h`.

Retrieves a vector of `CADIPParameterInfo_t` objects for all the parameters.



Use this function to get CADI parameter objects with all the relevant fields present for all EVSs, external SystemC modules, and loaded plug-ins.

```
SCX_API std::vector< eslapi::CADIPParameterInfo_t > scx::
scx_get_cadi_parameter_infos()
```



Plug-ins must be loaded before calling any of the platform parameter functions, otherwise their parameters will not be available. Any plug-in specified after the first call to any platform parameter function is ignored.

### **scx\_query\_cadi\_parameter\_infos()**

Defined in `scx_simconfig.h`.

Retrieves a vector of `CADIPParameterInfo_t` objects for all the parameters, including dependent parameters.

Use this function to get CADI parameter objects with all the relevant fields present for all EVSs, external SystemC modules, and loaded plug-ins. Any EVS parameters that are dependent on other parameters are also returned, but have invalid ID fields.

The dependent parameters are generated based on the parameters that have already been set using `scx_set_parameter()` and the extra parameter values passed as `paramValuePairs`. These extra parameter values do not alter the currently set parameters and are only used to infer what dependent parameters are available assuming they were set. The extra parameter values take precedence over any already set parameters.

Diagnostics for the generation of dependent parameters are written to `errorStream`.

```
SCX_API std::vector< eslapi::CADIPParameterInfo_t > scx::
scx_query_cadi_parameter_infos(std::ostream &errorStream, std::map< std::string,
std::string > const &paramValuePairs)
```

Parameters:

#### **errorStream**

The ostream for returning errors.

#### **paramValuePairs**

A `<string,string>` map for returning parameters and values.



Plug-ins must be loaded before calling any of the platform parameter functions, otherwise their parameters will not be available. Any plug-in specified after the first call to any platform parameter function is ignored.

**scx\_set\_cpi\_file()**

Defined in `scx_simconfig.h`.

Sets the Cycles Per Instruction (CPI) file for CPI class functionality.

Use this function to activate the CPI class functionality.

```
SCX_API void scx::scx_set_cpi_file(const std::string &cpi_file_path)
```

Parameters:

**cpi\_file\_path**

Specifies the path to the CPI file.



Note

This function must be called before any call to a platform parameter function.



Note

This function is deprecated and will be removed in a future release.

**scx\_cpulimit()**

Defined in `scx_simconfig.h`.

Sets the maximum number of CPU (User + System) seconds to run, excluding startup and shutdown.

```
SCX_API void scx::scx_cpulimit(double t)
```

Parameters:

**t**

The number of seconds to run. Defaults to unlimited.

**scx\_timelimit()**

Defined in `scx_simconfig.h`.

Sets the maximum number of seconds to run, excluding startup and shutdown.

```
SCX_API void scx::scx_timelimit(double t)
```

Parameters:

**t**

The number of seconds to run. Defaults to unlimited.

**scx\_add\_breakpoint()**

Defined in `scx_simconfig.h`.

Sets a breakpoint on the specified address.

```
SCX_API void scx::scx_add_breakpoint(std::string instance, uint64_t addr, const
std::string &memspace)
```

Parameters:

**addr**

Address at which to set the breakpoint.

**instance**

Name of the instance to set the breakpoint on.

**memspace**

Memory space on which to set the breakpoint.

**scx\_set\_start\_pc()**

Defined in `scx_simconfig.h`.

Sets the initial value of the PC register.

```
SCX_API void scx::scx_set_start_pc(std::string instance, uint64_t addr)
```

Parameters:

**addr**

Address to set the PC to.

**instance**

Name of the instance to set the PC on.

**scx\_dump()**

Defined in `scx_simconfig.h`.

Sets the details of a memory dump to be written to a file.

```
SCX_API void scx::scx_dump(std::string instance, std::string filename, std::string
memSpace, uint64_t addr, uint64_t size)
```

Parameters:

**addr**

Address to start dumping from.

**filename**

Filename to save the memory dump to.

**instance**

Name of the instance to dump memory from.

**memSpace**

Memory space name or ID.

**size**

Size in bytes of memory to dump.

**scx\_load\_params\_file()**

Defined in `scx_simconfig.h`.

Loads parameter values from the given configuration file.

```
SCX_API void scx::scx_load_params_file(const std::string &filename)
```

Parameters:

**filename**

Specifies the name of the configuration file to load.



Plug-ins must be loaded before calling any of the platform parameter functions, otherwise their parameters will not be available.

---

**scx\_list\_instances()**

Defined in `scx_simconfig.h`.

Lists all simulation instances.

```
SCX_API void scx::scx_list_instances(const std::string &filename=std::string())
```

Parameters:

**filename**

Specifies the path to the file for the output. The default is an empty string, which sends output to `std::cout`.

**scx\_dump\_instances()**

Defined in `scx_simconfig.h`.

Dumps all simulation instances in JSON format.

```
SCX_API void scx::scx_dump_instances(const std::string &filename=std::string())
```

Parameters:

**filename**

Specifies the path to the file for the output. Defaults to `instance_list.json` if filename is empty.

**scx\_list\_registers()**

Defined in `scx_simconfig.h`.

Lists all simulation registers.

```
SCX_API void scx::scx_list_registers(const std::string &filename=std::string())
```

Parameters:

**filename**

Specifies the path to the file for the output. The default is an empty string, which sends output to `std::cout`.

**scx\_list\_memory()**

Defined in `scx_simconfig.h`.

Lists all simulation memory.

```
SCX_API void scx::scx_list_memory(const std::string &filename=std::string())
```

Parameters:

**filename**

Specifies the path to the file for the output. The default is an empty string, which sends output to `std::cout`.

**scx\_parse\_and\_configure()**

Defined in `scx_simconfig.h`.

Parses command-line options and configures the simulation.

The application must pass the values of the options from function `sc_main()` as arguments to this function.

The supported options are the same as the command-line options for FVPs, see Fast Models FVPs Reference Guide ( <https://developer.arm.com/documentation/110379/latest/>).

Additionally, all remaining command-line arguments are treated as applications to load.

This function calls `std::exit(EXIT_SUCCESS)` to exit for options `--list-params` and `--help`. It calls `std::exit(EXIT_FAILURE)` in case of an error in a parameter specifier, an invalid option, or an application or plug-in was not found.

```
SCX_API void scx::scx_parse_and_configure(int argc, char *const argv[], const char *trailer=nullptr, bool sig_handler=true)
```

Parameters:

**argc**

Number of command-line options.

**argv**

Array of command-line options.

**sig\_handler**

Specifies whether to enable signal handler function. `true` to enable (default), `false` to disable.

**trailer**

Specifies a string that follows the option list when printing the help message ( `--help` option).

### **scx\_register\_synchronous\_thread()**

Defined in `scx_simconfig.h`.

Registers a new thread in the simulation engine which is implicitly synchronized with the simulation thread.

The caller must make sure that the simulation thread and the newly-registered thread do not run concurrently.

Calling this function for a thread completely disables the thread synchronization for that thread, that is, marshaling of function calls from the calling thread onto the simulation thread, for example Iris calls.

This function is useful for debugger threads that are blocking the simulation thread and still want to issue Iris calls while the simulation thread is blocked.

```
SCX_API void scx::scx_register_synchronous_thread(std::thread::id thread_id)
```

Parameters:

**thread\_id**

The id of the thread to be registered.

### **scx\_get\_error\_count()**

Defined in `scx_simconfig.h`.

Retrieves the number of errors recorded by the simulation engine.

```
SCX_API size_t scx::scx_get_error_count()
```



This function returns internal errors recorded by the simulation engine, some of which are not reported as errors by `scx_report_handler`.

### **scx\_get\_exitcode\_list()**

Defined in `scx_simconfig.h`.

Retrieves the list of exit codes that were logged by the simulation engine.

The returned list is a `std::vector` that contains the logged exit codes in order. Each entry in the list is a struct of type `scx::scx_exitcode_entry`. The last entry is the most recent.

```
SCX_API scx_exitcode_list_t scx::scx_get_exitcode_list()
```

Returns the exit code list.



If no exit code was recorded, the returned list is empty. This function only produces valid output after `sc_start()` has returned and it should not be called beforehand.

### **scx\_get\_default\_simcontrol()**

Defined in `scx_simcontrol.h`.

Returns a pointer to the default implementation of the simulation controller provided with Fast Models.

```
scx_simcontrol_if * scx::scx_get_default_simcontrol()
```

### **scx\_get\_curr\_simcontrol()**

Defined in `scx_simcontrol.h`.

Returns a pointer to the current implementation of the simulation controller.

```
SCX_API scx_simcontrol_if * scx::scx_get_curr_simcontrol()
```

### **scx\_set\_run\_simulation\_at\_start()**

Defined in `scx_simdebug.h`.

Specifies whether to run the simulation immediately at startup.

The default behavior is the simulation always runs at startup. However, if the user starts the Iris server by either using the `-I` or `--iris-server` command-line option or by calling `scx_enable_iris_server()`, the simulation is put into a wait state until the debugger starts it.

If the `-R` or `--run` command-line option is used, the simulation runs at startup.

```
SCX_API void scx::scx_set_run_simulation_at_start(bool run=true)
```

Parameters:

**run**

`true` (default) to immediately run the simulation, `false` otherwise.

### **scx\_set\_exit\_simulation\_at\_stop()**

Defined in `scx_simdebug.h`.

Specifies whether to exit the simulation when it stops.

The default behavior is the simulation always exits when it stops. However, if the user has started the Iris server, the simulation does not exit at stop state. Instead it waits for the debugger to either resume or shut down the simulation.

```
SCX_API void scx::scx_set_exit_simulation_at_stop(bool exit=true)
```

Parameters:

**exit**

`true` (default) to exit the simulation, `false` otherwise.

### **scx\_print\_port\_number()**

Defined in `scx_simdebug.h`.

Specifies whether to enable printing the TCP port number that the Iris server is listening to.

```
SCX_API void scx::scx_print_port_number(bool print=true)
```

Parameters:

**print**

`true` to enable printing the TCP port number, `false` otherwise.



Printing the TCP port number cannot be enabled after the simulation has started.



## **scx\_print\_statistics()**

Defined in `scx_simdebug.h`.

Specifies whether to enable printing simulation statistics at the end of the simulation.

```
SCX_API void scx::scx_print_statistics(bool print=true)
```

Parameters:

### **print**

`true` to enable printing simulation statistics, `false` otherwise.



Note

Printing statistics cannot be enabled after the simulation has started.



Note

The printed statistics include the startup time, which includes the run time for LISA `reset()` behaviors and the load time for the application. A long simulation run compensates for this.

---

## **scx\_load\_trace\_plugin()**

Defined in `scx_simdebug.h`.

This function is deprecated. Use `scx_load_plugin()` instead.

```
SCX_API void scx::scx_load_trace_plugin(const std::string &file)
```

Parameters:

### **file**

Specifies the path and filename of the trace plug-in to load.

## **scx\_load\_plugin()**

Defined in `scx_simdebug.h`.

Specifies a plug-in to be loaded.

Use this function to specify a plug-in to be loaded. It will be loaded at `end_of_elaboration()`, at the latest, or when any of the platform parameter functions is called.

```
SCX_API void scx::scx_load_plugin(const std::string &file)
```

Parameters:

**file**

Plug-in file to be loaded.

**Note**

Plug-ins must be loaded before calling any of the platform parameter functions, otherwise their parameters will not be available. Any plug-in loaded after the first call to any platform parameter function will be ignored.

**scx\_get\_global\_interface()**

Defined in `scx_simdebug.h`.

Requests a pointer to the global interface.

Use this function to access the global interface of the simulation, which allows access to all interfaces that are registered in the simulation.

```
SCX_API eslapi::CAInterface * scx::scx_get_global_interface()
```

Returns a pointer to the global interface, or `NULL` if not found.

## 5.6 scx\_simcallback\_if class

Defined in `scx_simcontrol.h`.

This interface is implemented by the simulation engine and used by the simulation controller to notify changes in the simulation state.

**notify\_running()**

Notifies that the simulation is running.

The simulation controller calls this function to notify debuggers that the simulation is running.

```
virtual void scx::scx_simcallback_if::notify_running()=0
```

**notify\_stopped()**

Notifies that the simulation has stopped.

The simulation controller calls this function to notify debuggers that the simulation has stopped.

```
virtual void scx::scx_simcallback_if::notify_stopped()=0
```

**notify\_debuggable()**

Notifies that the simulation is debuggable.

The simulation controller periodically calls this function, typically while the simulation is stopped, to notify that the simulation is debuggable. This allows clients to process debug activity, for instance memory or breakpoint operations.

This version of the function does nothing.

```
virtual void scx::scx_simcallback_if::notify_debuggable()
```

### **notify\_idle()**

Notifies that the simulation is idle.

The simulation controller periodically calls this function, typically while the simulation is stopped, to notify that the simulation is idle. This allows clients to process idle activity, for instance processing GUI events and redrawing.

```
virtual void scx::scx_simcallback_if::notify_idle()=0
```

### **~scx\_simcallback\_if()**

Destructor.

This version of the function does not allow destroying instance through this interface.

```
virtual scx::scx_simcallback_if::~scx_simcallback_if()
```

## 5.7 scx\_simcontrol\_if class

Defined in `scx_simcontrol.h`.

This interface must be implemented by the simulation controller, which interacts with the simulation engine.

This interface is used by the simulation engine to access the current implementations of the scheduler and report handler, as well as to request changes to the simulation state.

All simulation requests provided by this interface are asynchronous and are therefore expected to return immediately, whether the corresponding operation has completed or not. When the operation has completed, the corresponding notification must be sent back to the simulation that in turn notifies all connected debuggers accordingly to allow them to update their state.

Unless otherwise stated, an implementation of this interface must be thread-safe, that is it must not make assumptions about threads that issue simulation requests.

The default implementation of the simulation controller provided with Fast Models is at `$MAXCORE_HOME/lib/template/tpl_scx_simcontroller.{h,cpp}`.

### **get\_scheduler()**

Returns a pointer to the current implementation of the simulation scheduler.

This function is called by the simulation engine to retrieve the scheduler implementation for the simulation at construction time.

```
virtual eslapl::CAInterface * scx::scx_simcontrol_if::get_scheduler()=0
```



Note

An implementation of this function does not need to be thread-safe.

### **get\_report\_handler()**

Returns a pointer to the current implementation of the report handler.

This function is called by `scx_initialize()` to retrieve the report handler implementation for the simulation at construction time.

```
virtual scx_report_handler_if * scx::scx_simcontrol_if::get_report_handler()=0
```



Note

An implementation of this function does not need to be thread-safe.

### **run()**

Requests to run the simulation.

This function is called by the simulation engine upon receipt of an Iris run request from a debugger.

```
virtual void scx::scx_simcontrol_if::run()=0
```

### **stop()**

Requests to stop the simulation as soon as possible, that is at the next `wait()`.

This function is called by the simulation engine upon receipt of an Iris stop request from a debugger, a component, or when a breakpoint is hit.

```
virtual void scx::scx_simcontrol_if::stop()=0
```

### **is\_running()**

Returns whether the simulation is running.

This function is called by the simulation engine upon receipt of an Iris request from a debugger that needs to know whether the simulation is running.

```
virtual bool scx::scx_simcontrol_if::is_running()=0
```

Returns `true` if the simulation is running, `false` if it is paused or stopped.

### **stop\_acknowledge()**

Blocks the simulation while it is stopped.

This function is called by the scheduler thread to effectively stop the simulation, as a side-effect of calling `stop()` to request the simulation is stopped.

An implementation of this function must call `clearStopRequest()` on the given `runnable` (when not `NULL`).

```
virtual void scx::scx_simcontrol_if::stop_acknowledge(sg::SchedulerRunnable  
*runnable)=0
```

Parameters:

#### **runnable**

Specifies a pointer to the scheduler thread calling `stop_acknowledge()`.

### **process\_debuggable()**

Processes debug activity while the simulation is at a debuggable point.

This function is called by the scheduler thread whenever the simulation is at a debuggable point, to enable debug activity to be processed.

An implementation of this function might simply call `scx_simcallback_if::notify_debuggable()` on all registered clients.

This version of the function does nothing.

```
virtual void scx::scx_simcontrol_if::process_debuggable()
```

### **notify\_pending\_debug()**

Notifies the simcontroller that debug requests are pending and need processing as soon as possible while the simulation is stopped.

An implementation of this function might simply call `scx_simcontrol::process_debuggable()` on all registered clients, while the simulation is stopped in `scx_simcontrol::stop_acknowledge()`.

```
virtual void scx::scx_simcontrol_if::notify_pending_debug()
```

**process\_idle()**

Processes idle activity while the simulation is stopped.

This function is called by the scheduler thread whenever the simulation is idle to enable idle activity to be processed.

An implementation of this function might simply call `scx_simcallback_if::notify_idle()` on all registered clients.

```
virtual void scx::scx_simcontrol_if::process_idle()=0
```

**shutdown()**

Requests to shut down the simulation.

This function is called by the simulation engine to notify that it wants the simulation to shut down.

When the simulation has shut down, it cannot run again.

```
virtual void scx::scx_simcontrol_if::shutdown()=0
```



Note

There are no callbacks associated with this function.

---

**add\_callback()**

Registers callbacks with the simulation controller.

A client should call this function to register with the simulation controller a callback object that will handle notifications from the simulation.

```
virtual void scx::scx_simcontrol_if::add_callback(scx_simcallback_if  
*callback_obj)=0
```

Parameters:

**callback\_obj**

Specifies a pointer to the object whose member functions will be called as callbacks.

**remove\_callback()**

Removes callbacks from the simulation controller.

A client should call this function to remove any callback object it previously registered with the simulation.

```
virtual void scx::scx_simcontrol_if::remove_callback(scx_simcallback_if  
*callback_obj)=0
```

Parameters:

**callback\_obj**

Specifies a pointer to the object to remove.

**~scx\_simcontrol\_if()**

Destructor.

This version of the function does not allow destroying instances through this interface.

```
virtual scx::scx_simcontrol_if::~~scx_simcontrol_if()
```

## 6. Scheduler API

The Fast Models Scheduler API enables modeling components and systems in different environments, with or without a built-in scheduler. Examples are a SystemC environment or a standalone simulator.

The Fast Models Scheduler API is a C++ interface consisting of a set of abstract base classes. The header file that defines them is `$PVLIB_HOME/include/fmruntime/sg/SGSchedulerInterfaceForComponents.h`. This header file depends on other header files under `$PVLIB_HOME/include/`.

All Scheduler API constructs are in the namespace `sg`.

The interface decouples the modeling components from the scheduler implementation. The parts of the Scheduler API that the modeling components use are for the scheduler or scheduler adapter to implement. The parts that the scheduler or scheduler adapter use are for the modeling components to implement.

### 6.1 Intended mapping of the Scheduler API onto SystemC/TLM

This topic describes how Scheduler API functionality might map onto SystemC functionality.

**`sg::SchedulerInterfaceForComponents::wait(time)`**

Call `sc_core::wait(time)` and handle all pending asynchronous events that are scheduled with `sg::SchedulerInterfaceForComponents::addCallback()` before waiting.

**`sg::SchedulerInterfaceForComponents::wait(sg::ThreadSignal)`**

Call `sc_core::wait(sc_event)` on the `sc_event` in `sg::ThreadSignal` and handle all pending asynchronous events that are scheduled with `sg::SchedulerInterfaceForComponents::addCallback()` before waiting.

**`sg::SchedulerInterfaceForComponents::getCurrentSimulatedTime()`**

Return the current SystemC scheduler time in seconds as in `sc_core::sc_time_stamp().to_seconds()`.

**`sg::SchedulerInterfaceForComponents::addCallback()` and `removeCallback()`**

SystemC has no way to trigger simulation events from alien (non-SystemC) host threads in a thread-safe way. Buffer and handle these asynchronous events in all regularly re-occurring scheduler events. Handling regular simulation `wait()` and `timerCallback()` calls is sufficient.

**`sg::SchedulerInterfaceForComponents::stopRequest()` and `stopAcknowledge()`**

Pause and resume the SystemC scheduler. This function is out of scope of SystemC/TLM functionality, but in practice every debuggable SystemC implementation has ways to pause and resume the scheduler. Do not confuse these functions with `sc_core::sc_stop()`, which exits the SystemC simulation loop. They work with the `sg::SchedulerRunnable` instances and the `scx::scx_simcontrol_if` interface.



**sg::SchedulerInterfaceForComponents::createThread(), createThreadSignal(), createTimer()**

Map these functions onto SystemC threads created with `sc_spawn()` and `sc_events`. You can create and destroy `sg::SchedulerThread`, `sg::ThreadSignal`, and `sg::Timer` objects during elaboration, and delete them at runtime, unlike their SystemC counterparts. This process requires careful mapping. For example, consider what happens when you remove a waited-for `sc_event`.

**sg::ThreadSignal**

Map onto `sc_event`, which is notifiable and waitable.

**sg::SchedulerThread**

Map onto a SystemC thread that was spawned with `sc_core::sc_spawn()`. The thread function can call `sg::SchedulerThread::threadProc()`.

**sg::Timer**

Map onto a SystemC thread that, after the timer is `set()`, issues calls to the callbacks in the intervals (according to the `set()` interval).

## 6.2 Accessing SchedulerInterfaceForComponents from a modeling component

This topic shows ways of accessing the `SchedulerInterfaceForComponents` interface from a LISA, C++, and SystemC component.

- LISA component:

```
includes
{
    #include "sg/SGSchedulerInterfaceForComponents.h"
    #include "sg/SGComponentRegistry.h"
}

behavior init
{
    sg::SchedulerInterfaceForComponents *scheduler =
        sg::obtainComponentInterfacePointer<sg::SchedulerInterfaceForComponents>
            (getGlobalInterface(), "scheduler");
}
```

- C++ component:

```
#include "sg/SGSchedulerInterfaceForComponents.h"
#include "sg/SGComponentRegistry.h"

sg::SchedulerInterfaceForComponents *scheduler =
    sg::obtainComponentInterfacePointer<sg::SchedulerInterfaceForComponents>
        (simulationContext->getGlobalInterface(), "scheduler");
```

**Note**

C++ components have an `sg::SimulationContext` pointer passed into their constructor.

- SystemC component:

```
#include "sg/SGSchedulerInterfaceForComponents.h"
#include "sg/SGComponentRegistry.h"

sg::SchedulerInterfaceForComponents *scheduler =
    sg::obtainComponentInterfacePointer<sg::SchedulerInterfaceForComponents>
        (scx::scx_get_global_interface(), "scheduler");
```

## 6.3 FrequencyObserver class

Defined in `SGSchedulerInterfaceForComponents.h`.

Abstract base class for objects that want to receive notification messages from `FrequencySource` objects when the clock frequency changes.

### **notifyFrequencyChanged()**

Notifies this object that the observed frequency source has changed.

The object should query the new clock frequency using `frequencySource->getFrequency()` and adapt its behavior accordingly.

The object must not cause the `FrequencySource` to change its frequency again from within this callback.

```
virtual void
sg::vl_scheduler::FrequencyObserver::notifyFrequencyChanged(FrequencySource
    *frequencySource)=0
```

Parameters:

#### **frequencySource**

`FrequencySource` object that has changed and that issued this notification. This is the object that this `FrequencyObserver` previously registered using `registerFrequencyObserver()`. It is never `NULL`.

### **notifyFrequencySourceDestroyed()**

Notifies this object that the observed frequency source is about to be destroyed.

It is valid to access `frequencySource` from within this function.

It is valid, but not required, to call `frequencySource->unregisterFrequencyObserver()` from within this function. `frequencySource` can no longer be accessed after returning from this function.

```
virtual void sg::vl_scheduler::FrequencyObserver::
notifyFrequencySourceDestroyed(FrequencySource *frequencySource)=0
```

Parameters:

**frequencySource**

FrequencySource object that is about to be destroyed and that issues this notification. This is the object that this `FrequencyObserver` previously registered using `registerFrequencyObserver()`.

**~FrequencyObserver ()**

Protected virtual destructor. Prevents the destruction of instances through this interface.

```
sg::vl_scheduler::FrequencyObserver::~~FrequencyObserver() override
```

## 6.4 FrequencySource class

Defined in `sgSchedulerInterfaceForComponents.h`.

Provides a clock frequency. The clock frequency can be queried using `getFrequency()` and the object can notify other objects about clock frequency changes using `registerFrequencyObserver()` and `unregisterFrequencyObserver()`.

**registerFrequencyObserver ()**

Registers an observer that wants to be notified whenever this object changes.

This object must not call `FrequencyObserver::notifyFrequencyChanged()` from within this function. Calling this function with an already-registered observer or 0 is allowed and is ignored.

```
virtual void
sg::vl_scheduler::FrequencySource::registerFrequencyObserver (FrequencyObserver
*observer)=0
```

Parameters:

**observer**

FrequencyObserver to be registered.

**unregisterFrequencyObserver ()**

Unregisters an observer that no longer wants to be notified about changes.

This object must not call `FrequencyObserver::notifyFrequencyChanged()` from within this function. Calling this function with an unknown, already-unregistered, or 0 observer is allowed and is ignored.

```
virtual void  
sg::v1_scheduler::FrequencySource::unregisterFrequencyObserver(FrequencyObserver  
*observer)=0
```

Parameters:

**observer**

FrequencyObserver to be unregistered.

**getFrequency()**

Gets the frequency in hertz of this clock source.

```
virtual double sg::v1_scheduler::FrequencySource::getFrequency()=0
```

**getFrequencyOrInvalid()**

Gets the frequency in hertz of this clock source, or `INVALID_FREQUENCY` if not valid.

It is optional to implement this function.

```
virtual double sg::v1_scheduler::FrequencySource::getFrequencyOrInvalid()
```

**getClockTree()**

Gets as much of the clock tree as possible.

The return value is only valid immediately after this call and must be copied immediately.

It is optional to implement this function.

```
virtual char const * sg::v1_scheduler::FrequencySource::getClockTree()
```

**INVALID\_FREQUENCY()**

Returns a constant -1.0 number.

```
static double sg::v1_scheduler::FrequencySource::INVALID_FREQUENCY()
```

**notifyFrequencyObservers()**

Notifies all registered observers that this object has changed.

This function is not part of the public interface and is only an implementation guideline for implementers of this class.

```
virtual void sg::v1_scheduler::FrequencySource::notifyFrequencyObservers()=0
```

### **~FrequencySource ()**

Protected virtual destructor. Prevents the destruction of instances through this interface.

```
sg::v1_scheduler::FrequencySource::~FrequencySource() override
```

## 6.5 GlobalFrequencySource class

Defined in `sgSchedulerInterfaceForComponents.h`.

### **GlobalFrequencySource ()**

Constructor.

```
sg::v1_scheduler::SchedulerInterfaceForComponents::GlobalFrequencySource::  
GlobalFrequencySource(double f)
```

Parameters:

**f**

Frequency.

### **~GlobalFrequencySource ()**

Destructor.

```
sg::v1_scheduler::SchedulerInterfaceForComponents::GlobalFrequencySource::  
~GlobalFrequencySource() override
```

### **getName ()**

Returns the constant name `sg.GlobalFrequencySource`.

```
const char * sg::v1_scheduler::SchedulerInterfaceForComponents::  
GlobalFrequencySource::getName() const override
```

### **registerFrequencyObserver ()**

Registers a `FrequencyObserver`.

```
void sg::v1_scheduler::SchedulerInterfaceForComponents::GlobalFrequencySource::  
registerFrequencyObserver(FrequencyObserver *f) override
```

Parameters:

**f**

FrequencyObserver to be registered.

**unregisterFrequencyObserver()**

Unregisters a FrequencyObserver.

```
void sg::v1_scheduler::SchedulerInterfaceForComponents::GlobalFrequencySource::
unregisterFrequencyObserver(FrequencyObserver *f) override
```

Parameters:

**f**

FrequencyObserver to be unregistered.

**getFrequency()**

Returns the current frequency.

```
double sg::v1_scheduler::SchedulerInterfaceForComponents::
GlobalFrequencySource::getFrequency() override
```

**notifyFrequencyObservers()**

Triggers a notification to all registered observers.

```
void sg::v1_scheduler::SchedulerInterfaceForComponents::GlobalFrequencySource::
notifyFrequencyObservers() override
```

## 6.6 SchedulerCallback class

Defined in `sgSchedulerInterfaceForComponents.h`.

Callback base class for callbacks registered with

```
SchedulerInterfaceForComponents::addCallback() OR
SchedulerInterfaceForComponents::removeCallback().
```

**schedulerCallback()**

Callback function.

Always called on the simulation thread as soon as possible after the callback was registered with `SchedulerInterfaceForComponents::addCallback()`.

Callbacks automatically remove themselves after they were called. It is not necessary (nor allowed) to call `SchedulerInterfaceForComponents::removeCallback(this)` from within the callback. It is

not allowed to call `SchedulerInterfaceForComponents::addCallback()` or `removeCallback()` from within this callback.

```
virtual void sg::v1_scheduler::SchedulerCallback::schedulerCallback()=0
```

### **~SchedulerCallback()**

Protected virtual destructor. Prevents the destruction of instances through this interface.

```
sg::v1_scheduler::SchedulerCallback::~~SchedulerCallback() override
```

## 6.7 SchedulerInterfaceForComponents class

Defined in `sgSchedulerInterfaceForComponents.h`.

Used by modeling components to create threads and timers, for example.

This interface is not usually used by the owner of the scheduler. Every simulation framework must implement it and can use any kind of scheduler behind it.

Modeling components usually use the following function to access this interface:

```
sg::obtainComponentInterfacePointer<sg::SchedulerInterfaceForComponents>  
(  
    simulationContext->getGlobalInterface(),  
    "scheduler"  
)
```

### **IFNAME()**

Returns the constant name `sg.SchedulerInterfaceForComponents`.

```
static eslapl::if_name_t sg::v1_scheduler::SchedulerInterfaceForComponents::  
IFNAME()
```

### **IFREVISION()**

Returns the constant version number 1.

```
static eslapl::if_rev_t sg::v1_scheduler::SchedulerInterfaceForComponents::  
IFREVISION()
```

### **createTimer()**

Creates a new `Timer` object with the specified name and callback object.

When the `Timer` fires, `TimerCallback::timerCallback()` is called. If `callback` is `NULL`, this call returns `NULL` and is ignored. This function is used by `clockTimer` and `clockTimer64`.

```
virtual Timer * sg::v1_scheduler::SchedulerInterfaceForComponents::createTimer(const
char *name, TimerCallback *callback)=0
```

Parameters:

**callback**

Timer callback function, called when the timer expires.

**name**

The instance name of the timer. If `name` is `NULL`, it is given the name `(anonymous timer)`. This function makes a copy of `name`.

Returns an object implementing `timer`. It is only ever `NULL` when `callback == 0`.

**createThreadSignal()**

Creates a thread signal.

A thread signal is a non-schedulable but waitable event on which many threads may wait. When the event is signaled, all threads that wait on this event are scheduled to run. Destroying the returned object while threads are waiting for it leaves the threads unscheduled. This function is used by CT cores.

```
virtual ThreadSignal * sg::v1_scheduler::SchedulerInterfaceForComponents::
createThreadSignal(const char *name)=0
```

Parameters:

**name**

The instance name of the thread signal. Ideally, the hierarchical name of the modeling component that owns the thread is included in the name. If `name` is `NULL`, it is given the name `(anonymous thread signal)`. This function makes a copy of `name`.

Returns a pointer to an object implementing the `ThreadSignal`.

**wait()**

The `wait()` function is overloaded. These are the different versions of this function:

- ```
virtual void sg::v1_scheduler::SchedulerInterfaceForComponents:: wait(ticks_t
ticks)
```

Blocks the current thread and runs other threads for some time.

Calling this function from outside of a `SchedulerRunnable::threadProc()` context is invalid.

This function blocks the currently-running thread for the specified amount of time and lets other threads run. It returns when the calling thread should continue to run. This is the coroutine switching point.



Typically, a thread calls `wait(n)` in its loop each time it completes `n` ticks of work. `n` is called a quantum.

`ticks` may be 0. This means that the current thread is potentially blocked and if any other callbacks are pending, they are run, but no time is advanced for this thread and this thread is probably scheduled again immediately.

Parameters:

**ticks**

Time to wait for, relative to simulated time resolution.

- ```
virtual void sg::vl_scheduler::SchedulerInterfaceForComponents:: wait(ticks_t ticks, FrequencySource *timebase)
```

Blocks the current thread and runs other threads for some time.

A timebase of NULL is valid and results in this call being ignored. A timebase frequency of 0.0 is valid and results in this call being ignored. The time to wait for is calculated as `ticks / timebase->getFrequency()`.

Parameters:

**ticks**

Time to wait for. This is relative to the frequency specified by `timebase`.

**timebase**

Specifies how to interpret `ticks` in terms of time.



Note

This function is deprecated and will be removed in a future release.

- ```
virtual void sg::vl_scheduler::SchedulerInterfaceForComponents:: wait(ThreadSignal *threadSignal)=0
```

Waits on a thread signal.

Blocks the current thread indefinitely until `threadSignal->notify()` is called.

This function should be called from within a `SchedulerRunnable::threadProc()` context. Calling it from a context outside of a `threadProc()` is valid and has no effect. Calling it with a NULL `threadSignal` has no effect.

This function returns as soon as the calling thread can continue to run.

Parameters:

**threadSignal**

Thread signal object that should be waited on.

## setGlobalQuantum()

The `setGlobalQuantum()` function is overloaded. These are the different versions of this function:

- ```
virtual void sg::v1_scheduler::SchedulerInterfaceForComponents::
    setGlobalQuantum(ticks_t ticks)
```

Sets the value of the global quantum.

The global quantum is the maximum time a thread can run ahead of simulation time. All threads must synchronize on timing points that are multiples of the global quantum.

Parameters:

### **ticks**

Global quantum value measured in ticks relative to simulated time resolution.

- ```
virtual void sg::v1_scheduler::SchedulerInterfaceForComponents::
    setGlobalQuantum(ticks_t ticks, FrequencySource *timebase)
```

Sets the value of the global quantum.

The global quantum is the maximum time a thread can run ahead of simulation time. All threads must synchronize on timing points that are multiples of the global quantum.

Parameters:

### **ticks**

Global quantum value measured in `timebase` units. The value is calculated as `ticks / timebase->getFrequency()`.

### **timebase**

Specifies how to interpret `ticks` in terms of time.



Note

This function is deprecated and will be removed in a future release.

## getGlobalQuantum()

The `getGlobalQuantum()` function is overloaded. These are the different versions of this function:

- ```
virtual ticks_t sg::v1_scheduler::SchedulerInterfaceForComponents::
    getGlobalQuantum(Tag< ticks_t > *)
```

Returns the value of the global quantum in ticks relative to simulated time resolution.

- ```
virtual double sg::v1_scheduler::SchedulerInterfaceForComponents::
    getGlobalQuantum()
```

Returns the value of the global quantum in seconds.



This function is deprecated and will be removed in a future release.

## setMinSyncLatency()

The `setMinSyncLatency()` function is overloaded. These are the different versions of this function:

- ```
virtual void sg::v1_scheduler::SchedulerInterfaceForComponents::
  setMinSyncLatency(ticks_t ticks)
```

Sets the minimum synchronization latency for this scheduler.

The minimum synchronization latency helps to ensure that sufficient simulated time has passed between two synchronization points for synchronization to be efficient.

A small latency increases accuracy but decreases simulation speed. A large latency decreases accuracy but increases simulation speed.

The scheduler uses this information to compute the next synchronization point as returned by `getNextSyncPoint()`.

Parameters:

### **ticks**

Specifies the minimum synchronization latency measured in ticks relative to simulated time resolution.

- ```
virtual void sg::v1_scheduler::SchedulerInterfaceForComponents::
  setMinSyncLatency(ticks_t ticks, FrequencySource *timebase)
```

Sets the minimum synchronization latency for this scheduler.

The minimum synchronization latency helps to ensure that sufficient simulated time has passed between two synchronization points for synchronization to be efficient.

A small latency increases accuracy but decreases simulation speed. A large latency decreases accuracy but increases simulation speed.

The scheduler uses this information to compute the next synchronization point as returned by `getNextSyncPoint()`.

Parameters:

### **ticks**

Specifies the minimum synchronization latency measured in `timebase` units. The value should be calculated as `ticks / timebase->getFrequency()`.

### **timebase**

Specifies how to interpret `ticks` in terms of time.



This function is deprecated and will be removed in a future release.

### **getMinSyncLatency()**

The `getMinSyncLatency()` function is overloaded. These are the different versions of this function:

- ```
virtual ticks_t sg::v1_scheduler::SchedulerInterfaceForComponents::
  getMinSyncLatency(Tag< ticks_t > *)
```

Returns the minimum synchronization latency for this scheduler, measured in ticks relative to simulated time resolution.

- ```
virtual double sg::v1_scheduler::SchedulerInterfaceForComponents::
  getMinSyncLatency()
```

Returns the minimum synchronization latency for this scheduler, measured in seconds.



This function is deprecated and will be removed in a future release.

### **addSynchronisationPoint()**

The `addSynchronisationPoint()` function is overloaded. These are the different versions of this function:

- ```
virtual void sg::v1_scheduler::SchedulerInterfaceForComponents::
  addSynchronisationPoint(ticks_t ticks)
```

Adds a future global synchronization point in time.

Modeling components can call this function to provide a hint to the scheduler implementation when a potentially useful global synchronization point will occur in the future. The scheduler uses this information to determine the quantum sizes of threads as they are scheduled.

Calling this function again adds another global synchronization point. Synchronization points are automatically removed when the time reaches them.

Parameters:

#### **ticks**

Time relative to the start of the current quantum (for threads) or relative to the current scheduler time (now) for all other callbacks. Measured in ticks relative to simulated time resolution.

**Note**

This function is deprecated and will be removed in a future release.

- ```
virtual void sg::v1_scheduler::SchedulerInterfaceForComponents::
  addSynchronisationPoint(ticks_t ticks, FrequencySource *timebase)
```

Adds a future global synchronization point in time.

Modeling components can call this function to provide a hint to the scheduler implementation when a potentially useful global synchronization point will occur in the future. The scheduler uses this information to determine the quantum sizes of threads as they are scheduled.

Calling this function again adds another global synchronization point. Synchronization points are automatically removed when the time reaches them.

Parameters:

**ticks**

Time relative to the start of the current quantum (for threads) or relative to the current scheduler time (now) for all other callbacks. Measured in `timebase` units. The time in the future should be calculated as `ticks / timebase->getFrequency()`.

**timebase**

Specifies how to interpret `ticks` in terms of time.

**Note**

This function is deprecated and will be removed in a future release.

## **getNextSyncPoint()**

The `getNextSyncPoint()` function is overloaded. These are the different versions of this function:

- ```
virtual ticks_t sg::v1_scheduler::SchedulerInterfaceForComponents::
  getNextSyncPoint(Tag< ticks_t > *)
```

Returns the next future global synchronization point in ticks relative to simulated time resolution.

Modeling components can call this function to get a hint for when a potentially useful global synchronization point will occur in the future. The quantum keeper uses this information to determine when core threads need to synchronize.

- ```
virtual double sg::v1_scheduler::SchedulerInterfaceForComponents::
  getNextSyncPoint()
```

Returns the next future global synchronization point in seconds relative to the current simulated time.

Modeling components can call this function to get a hint for when a potentially useful global synchronization point will occur in the future. The quantum keeper uses this information to determine when core threads need to synchronize.



Note

This function is deprecated and will be removed in a future release.

### getNextSyncRange ()

The `getNextSyncRange ()` function is overloaded. These are the different versions of this function:

- ```
virtual void sg::v1_scheduler::SchedulerInterfaceForComponents::
getNextSyncRange(double &min_sync_latency, double &next_sync_pt)
```

Returns the next synchronization range, which is [ minimum synchronization latency, next synchronization point ].

Modeling components can call this function to get a hint for when a potentially useful global synchronization point will occur in the future, and to update their view of the minimum synchronization latency. The core quantum keeper uses this information to determine when core threads need to and can synchronize.

Parameters:

**min\_sync\_latency**

Minimum synchronization latency, measured in seconds.

**next\_sync\_pt**

Next global synchronization point, measured in seconds relative to the current simulated time.



Note

This function is deprecated and will be removed in a future release.

- ```
virtual void sg::v1_scheduler::SchedulerInterfaceForComponents::
getNextSyncRange(ticks_t &min_sync_latency, ticks_t &next_sync_pt)
```

Returns the next synchronization range, which is [ minimum synchronization latency, next synchronization point ].

Modeling components can call this function to get a hint for when a potentially useful global synchronization point will occur in the future, and to update their view of the minimum

synchronization latency. The core quantum keeper uses this information to determine when core threads need to and can synchronize.

Parameters:

**min\_sync\_latency**

Minimum synchronization latency, measured in ticks relative to simulated time resolution.

**next\_sync\_pt**

Next global synchronization point, measured in ticks relative to simulated time resolution.

## **addCallback()**

Schedules a callback in the simulation thread.

This method may be called from any host thread and is thread-safe.

The callback function `callback->schedulerCallback()` is always called from the simulation thread, in other words the host thread that runs the simulation. It is called as soon as the scheduler has a chance to respond to the `addCallback()` function.

Multiple callbacks may be pending at any point in time. The scheduler can call these callbacks in any order.

It is not allowed to call `addCallback()` or `removeCallback()` from within the callback function.

A callback is automatically removed after it was called. It is generally not necessary to remove callbacks unless a pending callback which was not yet called should be canceled, for example when the object implementing the callback function is destroyed.

This method is used by `AsyncSignal`.

```
virtual void  
sg::vl_scheduler::SchedulerInterfaceForComponents::addCallback(SchedulerCallback  
*callback)=0
```

Parameters:

**callback**

The callback object to call. If NULL, this call is ignored.

## **removeCallback()**

Removes all callbacks scheduled using `addCallback()` for this callback object.

This method may be called from any host thread and is thread-safe.

If `callback` is NULL or is an unknown callback object or a callback that was already called (or is currently being called), this call is ignored.

The specified callback will no longer be called after this function returns. It may, however, be called while execution control is inside this function.

It is generally not necessary to remove callbacks unless a pending callback which was not yet called should be canceled, for example when the object implementing the callback function is destroyed.

This method is used by `AsyncSignal`.

```
virtual void
sg::v1_scheduler::SchedulerInterfaceForComponents::removeCallback(SchedulerCallback
*callback)=0
```

Parameters:

**callback**

The callback object to remove.

**getCurrentSimulatedTime()**

The `getCurrentSimulatedTime()` function is overloaded. These are the different versions of this function:

- ```
virtual ticks_t sg::v1_scheduler::SchedulerInterfaceForComponents::
getCurrentSimulatedTime(Tag< ticks_t > *)
```

Gets the current elapsed simulated time in seconds since the scheduler was created.

The granularity of this clock is determined by the timers and their expiry intervals and by the intervals of the `wait()` statements. It always accurately reflects the time of the last timer callback invocation or the last return from `SchedulerThread::wait()`, whichever occurred last.

The return value is guaranteed to monotonically increase over (real or simulated) time.

This method is used by `clockDivider` and `MasterClock (ClockSignalProtocol::currentTicks())`.

Returns the current simulated time in ticks, relative to simulated time resolution.

- ```
virtual double sg::v1_scheduler::SchedulerInterfaceForComponents::
getCurrentSimulatedTime()
```

Gets the current elapsed simulated time in seconds since the scheduler was created.

The granularity of this clock is determined by the timers and their expiry intervals and by the intervals of the `wait()` statements. It always accurately reflects the time of the last timer callback invocation or the last return from `SchedulerThread::wait()`, whichever occurred last.

The return value is guaranteed to monotonically increase over (real or simulated) time.

This method is used by `clockDivider` and `MasterClock (ClockSignalProtocol::currentTicks())`.

Returns the current simulated time in seconds.



**Note**

This function is deprecated and will be removed in a future release.

---

### **getSimulatedTimeResolution()**

Returns the simulated time resolution in seconds.

```
virtual double sg::v1_scheduler::SchedulerInterfaceForComponents::  
getSimulatedTimeResolution()
```

### **setSimulatedTimeResolution()**

Sets the simulated time resolution in seconds.

```
virtual void sg::v1_scheduler::SchedulerInterfaceForComponents::  
setSimulatedTimeResolution(double resolution)
```

Parameters:

#### **resolution**

Simulated time resolution in seconds.

---

**Note**

It is not possible to set the simulated time resolution after the simulation has started or after the timers have been set.

---

### **currentThread()**

Returns the current schedulerThread object or NULL if not in any threadProc() call.

```
virtual SchedulerThread * sg::v1_scheduler::SchedulerInterfaceForComponents::  
currentThread()
```

### **createThread()**

Creates a new thread.

See also the related classes schedulerThread and SchedulerRunnable.

This function does not yet start the new thread. Call schedulerThread::start() to start the new thread.

Destroying the returned object may or may not kill the thread, see the schedulerThread destructor.

This function is used by CT cores.

```
virtual SchedulerThread * sg::v1_scheduler::SchedulerInterfaceForComponents::
createThread(const char *name, SchedulerRunnable *runnable)=0
```

Parameters:

**name**

The instance name of the thread. Ideally, the hierarchical name of the modeling component that owns the thread is included in the name. If `name` is NULL, it is given the name (anonymous thread). This function makes a copy of `name`.

**runnable**

Object that implements the `SchedulerRunnable` interface. This is the object that contains the actual thread functionality. The returned thread uses this interface to communicate with the thread implementation in the modeling component. If NULL, this call returns NULL and is ignored.

Returns an object implementing `SchedulerThread`. It is never NULL except when `runnable` is NULL.

**stopRequest()**

Stop (pause) request. Requests the simulation of the whole system to pause.

Modeling components use this function to stop the simulation from within the simulation thread, for example breakpoints. Also, debuggers call it asynchronously from the debugger thread. It may be called from any host thread, and may be called while the simulation is running, which is the purpose of this function.

This function always immediately returns before the simulation is stopped. It does not block the caller until the simulation is stopped. The simulation stops as soon as possible in response to this call, depending on the `syncLevels` of the threads in the system.

In response to this call, the simulation calls the function `stopAcknowledge()` which in turn must block the simulation thread while the simulation should be paused. This function must not call `stopAcknowledge()` directly but should only set up a state such that the simulation knows it needs to stop at the next sync point, as defined by the `syncLevels` in the system. This state should later be reset by the `stopAcknowledge()` function calling `SchedulerRunnable::clearStopRequest()`. Calling this function again, from any host thread, before `stopAcknowledge()` has reset the stop request using `SchedulerRunnable::clearStopRequest()` is harmless and only stops the simulation once.

A `stopRequest()` is sufficient, but not necessary, to stop the simulation. The simulation may also stop, that is, call `stopAcknowledge()`, spontaneously, without a previous `stopRequest()`. This happens for example when a modeling component hits a breakpoint.

The scheduler implementation of this function should generally try to forward this `stopRequest()` to the currently-running runnable, see `SchedulerRunnable::stopRequest()`. It should only do this if `stopRequest()` is called from the simulation thread.

If the runnable accepts the `stopRequest()`, which is indicated by `SchedulerRunnable::stopRequest()` returning true, the scheduler does not need to do anything extra as the runnable will eventually respond with a `stopAcknowledge()` call. If the runnable does not accept the `stopRequest()`, which is indicated by `SchedulerRunnable::stopRequest()` returning false, or if this function is called outside of the context of any runnable, for example from a callback function or from a non-simulation host thread, the scheduler is responsible for handling the `stopRequest()` itself by calling `stopAcknowledge()` as soon as possible.

The entire stop-handling mechanism should not change the scheduling order or model behavior in any way, for non-intrusive debugging.

This function is used by CT cores, peripherals, and debuggers.

```
virtual void sg::v1_scheduler::SchedulerInterfaceForComponents::stopRequest()=0
```

### **stopAcknowledge()**

Blocks the simulation thread until it is told to resume.

This function is always called from within the simulation thread, either in response to a call to `stopRequest()`, or spontaneously, for example when a breakpoint is hit, or at a debugger stop.

The scheduler must block inside this function. The scheduler usually implements a thread-safe mechanism in this function which allows the simulation thread to be blocked and resumed from another host thread, usually the debugger thread. Calling this function from a non-simulation host thread is wrong by design and is not allowed.

This function must clear the stop request that led to calling this function by calling `runnable->clearStopRequest()`. This function must return as soon as the simulation should be resumed. It should not have any side effects except for blocking the simulation thread.

It is used by CT cores.

```
virtual void sg::v1_scheduler::SchedulerInterfaceForComponents::
stopAcknowledge(SchedulerRunnable *runnable)=0
```

Parameters:

#### **runnable**

Pointer to the runnable instance that called this function or NULL if not called from a runnable. If not NULL, this function calls `runnable->clearStopRequest()` when it is safe to do so, with respect to non-simulation host threads.

## 6.8 SchedulerObject class

Defined in `sgSchedulerInterfaceForComponents.h`.

Base class for all scheduler objects and interfaces.

**getName()**

Gets the name of the instance that implements this interface.

This name is not guaranteed to be unique or hierarchical, but we recommend including or using the hierarchical component name. It is intended for debugging purposes.

The caller must not free or delete the returned string. The string is owned by this object. The pointer is valid as long as the object implementing this interface exists. If the caller cannot track the lifetime of this object and wants to remember the name, they must make a copy of it.

```
virtual const char * sg::vl_scheduler::SchedulerObject::getName() const =0
```

Returns the instance name of this object.

**~SchedulerObject()**

Protected virtual destructor. Prevents the destruction of instances through this interface.

```
virtual sg::vl_scheduler::SchedulerObject::~~SchedulerObject()
```

## 6.9 SchedulerRunnable class

Defined in `sgSchedulerInterfaceForComponents.h`.

Threads interface (runnable side). These objects are created and implemented by the modeling components and a pointer to this interface is passed to `SchedulerInterfaceForComponents::createThread()`. The scheduler uses this interface to run the thread and control the thread execution.



Note

The `breakQuantum()` function is deprecated.

**threadProc()**

Main thread function. Thread entry point.

When this function returns, the thread will no longer be run and `threadProc()` will never be called again for this `schedulerThread` instance.

The thread usually does not return from this function while the thread is running.

`threadProc()` should call `SchedulerInterfaceForComponents::wait(0, ...)` after it completes its initialization code.

`threadProc()` should call `SchedulerInterfaceForComponents::wait(t>=0, ...)` after it completes `t` ticks worth of work.

```
virtual void sg::v1_scheduler::SchedulerRunnable::threadProc()=0
```

### **initThread()**

Thread initialization function. Called before `threadProc()`.

```
virtual void sg::v1_scheduler::SchedulerRunnable::initThread()
```

### **breakQuantum()**

Breaks the current quantum in `ticks` time, relative to the start of the quantum, or soon after that.

```
virtual void sg::v1_scheduler::SchedulerRunnable::breakQuantum(ticks_t ticks=0)
```

Parameters:

#### **ticks**

Specifies the number of ticks before breaking the quantum.



This function is deprecated and will be removed in a future release.

---

### **stopRequest()**

Stop (pause) request. Sets the stop request flag.

Requests the simulation of the whole system to pause as soon as possible such that this runnable is in a useful state for inspection through a debugger for example. For a CPU component, this might be at an instruction boundary.

This function may be called from any host thread. It may be called while the simulation is running, which is the purpose of this function.

This function always immediately returns before the simulation is stopped. It does not block the caller until the simulation is stopped. The simulation stops as soon as possible in response to this call, depending on the `syncLevel` of this runnable.

In response to this call, the simulation calls the function

`SchedulerInterfaceForComponents::stopAcknowledge()` which in turn must block the simulation thread while the simulation should be paused. This function must not call `stopAcknowledge()` directly but should only set up a state such that the simulation knows it needs to stop at the next sync point, as defined by the `syncLevel` of this runnable. This state should be reset by the `stopAcknowledge()` function by calling `clearStopRequest()`.

This function is used by modeling components to stop the simulation from within the simulation thread, for example external breakpoints, and also asynchronously from debuggers from the debugger thread.

Calling this function again, from any host thread, before `stopAcknowledge()` has reset the stop request using `SchedulerRunnable::clearStopRequest()` is harmless and only stops the simulation once.

Returns true if the runnable accepts the stop request and will stop later on. Returns false if the runnable does not accept the stop request. In this case, the scheduler must stop the simulation when the runnable returns control to the scheduler, using `wait()` for example.

```
virtual bool sg::v1_scheduler::SchedulerRunnable::stopRequest()=0
```

### **clearStopRequest()**

Clears the stop request flag.

This function is only called from `schedulerInterfaceForComponents::stopAcknowledge()`. This means it is always called from the simulation thread. It must clear the stop request flag.

```
virtual void sg::v1_scheduler::SchedulerRunnable::clearStopRequest()=0
```

### **setThreadProperty()**

Sets specific properties of the thread.

```
virtual bool sg::v1_scheduler::SchedulerRunnable::setThreadProperty(ThreadProperty  
property, uint64_t value)=0
```

Parameters:

**property**

Property to set.

**value**

Value the property is set to.

### **getThreadProperty()**

Gets the properties of the thread.

```
virtual bool sg::v1_scheduler::SchedulerRunnable::getThreadProperty(ThreadProperty  
property, uint64_t &valueOut)=0
```

Parameters:

**property**

Property to get.

**valueOut**

Value of the property.

**~SchedulerRunnable()**

Protected virtual destructor. Prevents the destruction of instances through this interface.

Modeling components control the lifetime of this object. It must live at least as long as the corresponding `schedulerThread` object exists.

```
sg::v1_scheduler::SchedulerRunnable::~~SchedulerRunnable() override
```

## 6.10 SchedulerRunnableWithGetRunnableName class

Defined in `sgSchedulerInterfaceForComponents.h`.

Modified version of `schedulerRunnable` interface class, with the `getName()` function renamed to `getRunnableName()`. The purpose of this class is to disambiguate it from other classes that might have a `getName()` method with a different prototype or different semantics which then defeats multiple inheritance.

The rest of the interface and semantics of this class are identical to class `schedulerRunnable`.

**getRunnableName()**

Publish renamed method.

```
virtual const char * sg::v1_scheduler::SchedulerRunnableWithGetRunnableName::  
getRunnableName() const =0
```

## 6.11 SchedulerThread class

Defined in `sgSchedulerInterfaceForComponents.h`.

Threads interface (thread instance/scheduler side). These objects are created using the `SchedulerInterfaceForComponents::createThread()` function. The modeling component uses this interface to communicate with the scheduler.



The `setFrequency()` function is deprecated and is not used.

---

### **~SchedulerThread()**

Destructor. May or may not kill the thread, depending on whether the underlying scheduler implementation supports killing threads or not.

The only clean way to end a thread is to signal it to return from its `threadProc()` function, for example by using an exception which is caught in the `threadProc()` function.

Destroying this object before `start()` was called should never start the thread. Destroying this object after `start()` was called may kill the thread immediately or may leave the thread running until it returns from its `threadProc()`.

```
sg::v1_scheduler::SchedulerThread::~SchedulerThread() override
```



Note

Killing threads without their cooperation is always unclean as it may leak resources.

### **setFrequency()**

Sets or resets the specified frequency source to be the parent clock for this thread.

This clock is used to interpret the `ticks` parameter of `breakQuantum()`. Specifying 0 is valid and is ignored. This object must register and unregister a `FrequencyObserver` to observe and respond to any frequency changes.

```
virtual void sg::v1_scheduler::SchedulerThread::setFrequency(FrequencySource  
*frequencySource)=0
```

Parameters:

#### **frequencySource**

A pointer to the new frequency source.



Note

This function is deprecated and will be removed in a future release.

### **start()**

Starts the thread.

This function calls `threadProc()` immediately which in turn must call `wait(0, ...)` after the initialization has completed in order for `start()` to return.

This function only runs the `threadProc()` function of the associated thread, not of any other threads.



Calling this function on a thread that has already started has no effect. Calling this function on a thread that has already terminated, in other words after `threadProc()` has returned, has no effect.

```
virtual void sg::v1_scheduler::SchedulerThread::start()=0
```

### **getRunnable()**

Returns the `SchedulerRunnable` of this thread.

The runnable might be used to access `ThreadProperties`.

```
virtual SchedulerRunnable * sg::v1_scheduler::SchedulerThread::getRunnable() const
```

## 6.12 Tag class

Defined in `SGSchedulerInterfaceForComponents.h`.

Used to select the v1 or v0 versions of the `get...()` methods of the `SchedulerInterfaceForComponents` class. Select the v1 version by passing a null pointer to the extra `Tag<>` argument.

## 6.13 ThreadSignal class

Defined in `SGSchedulerInterfaceForComponents.h`.

A non-schedulable but waitable event on which many threads may wait. When the event is signaled, all waiting threads are scheduled to run.

### **~ThreadSignal()**

Use the destructor to destroy thread signals.

Destroying this object while threads are waiting for it leaves the threads unscheduled.

```
sg::v1_scheduler::ThreadSignal::~~ThreadSignal() override
```

### **notify()**

Notify the event, waking up any waiting threads.

This function may be called from any `threadProc()` and also from outside `threadProc()`. Calling this function when no thread is waiting on this signal is allowed, but this call is ignored.

```
virtual void sg::v1_scheduler::ThreadSignal::notify()=0
```

## 6.14 Timer class

Defined in `sgschedulerInterfaceForComponents.h`.

Interface of timers. Instances of objects implementing this interface are created using the `createTimer()` method of `SchedulerInterfaceForComponents`.



The `setFrequency()` function is deprecated and is not used.



Values passed to `set()` represent ticks relative to simulated time resolution.



Values returned by `remaining()` represent ticks relative to simulated time resolution.

---

### **setFrequency()**

Sets the specified frequency source clock for this timer.

This function sets the initial frequency source for this timer or overrides the existing one.

Setting a new frequency source must not alter the state of the timer as much as possible. In particular, when changing the frequency, the timer should not be restarted and the remaining ticks (not the time) until the timer expires should be preserved.

Specifying a NULL frequency source is valid and cancels the timer.

This object must register and unregister a `FrequencyObserver` to observe and respond to any frequency changes.

```
virtual void sg::v1_scheduler::Timer::setFrequency(FrequencySource
*frequencySource)=0
```

Parameters:

#### **frequencySource**

New or initial frequency source for this timer.

**Note**

This function is deprecated and will be removed in a future release.

---

### **cancel()**

Unsets the timer, causing it to no longer fire.

Has no effect if the timer is not currently set.

```
virtual void sg::v1_scheduler::Timer::cancel()=0
```

### **set()**

Sets a timer to be signaled after `ticks` ticks of the clock.

When this happens, the user callback function is called. If the user callback returns 0, the timer acts as a one-shot, otherwise it will reoccur after `n` ticks, where `n` is the callback return value.

If the timer was already set and has not yet expired, it is reset as if the previous `set()` had never occurred.

```
virtual bool sg::v1_scheduler::Timer::set(ticks_t ticks)=0
```

Parameters:

#### **ticks**

The number of ticks in the future to set the timer to fire.

Returns false if the timer could not be scheduled. This can occur if `ticks` is too large.

### **isSet()**

Returns whether a timer is set and queued for callback.

This function is free of side effects.

```
virtual bool sg::v1_scheduler::Timer::isSet()=0
```

Returns true if the timer is set, false if not.

### **remaining()**

Gets the remaining clock ticks before a timer will be triggered.

This function may return 0 if the timer is about to be triggered. It returns 0 if the timer is not set.

This function is free of side effects.

```
virtual ticks_t sg::v1_scheduler::Timer::remaining()=0
```

Returns the number of remaining ticks before the timer will be triggered.

## 6.15 TimerCallback class

Defined in `sgSchedulerInterfaceForComponents.h`.

Callback base class for timers created through  
`SchedulerInterfaceForComponents::createTimer()`.

### **timerCallback()**

Timer callback function.

Called whenever the timer expires.

```
virtual ticks_t sg::v1_scheduler::TimerCallback::timerCallback()=0
```

Returns 0 for a one-shot timer or t to be signaled again in t ticks time.

### **~TimerCallback()**

Protected virtual destructor. Prevents the destruction of instances through this interface.

```
sg::v1_scheduler::TimerCallback::~TimerCallback() override
```

## 7. FastRAM

FastRAM is a fast interface to simulated RAM that lets platform models avoid using bus models for most transactions.

FastRAM uses a cache of DMI pointers to 64 MB regions. It tightly couples these DMI regions to the Fast Models bus masters and to models of IP that are bus masters. When FastRAM is enabled, accesses by Fast Models bus masters to platform RAM components bypass the PVBUS and TLM bus models. Accesses to other platform components and areas of RAM not covered by FastRAM continue to use normal bus paths.

FastRAM can provide significant speed improvements for large, complex platform models which can spend a lot of time in the bus models. It can particularly benefit SystemC platforms that use TLM, and multi-threaded platforms.

The behavior of platform models is functionally equivalent whether FastRAM is enabled or disabled. However, modeling bus transactions in a platform can lead to scheduling changes, so the overall flow of execution by components in a platform might not be identical.

Most platform models can safely use FastRAM, but there are conditions that can prevent its use, see [FastRAM limitations](#).

### 7.1 How to enable FastRAM

Enable FastRAM by launching the platform model with the command-line parameter `--fast-ram <config_file>`.

`config_file` is an ASCII file located in the current working directory of the platform model. It specifies:

- The physical address ranges to enable for FastRAM.
- Details of any address aliasing for the enabled ranges.
- Which bus masters to enable to use FastRAM.

For more information, see:

- [FastRAM configuration file syntax](#)
- [FastRAM configuration file example](#)
- [FastRAM limitations](#)

## 7.2 FastRAM configuration file syntax

The configuration file controls tracing, master selection, address ranges, aliasing, and comments.

Each line in the configuration file starts with a single character option followed by the required arguments, separated with whitespace.

**Table 7-1: FastRAM configuration options**

Option	Arguments	Description
T	-	Enable FastRAM trace on stdout from this point in the file.
Q	-	Disable FastRAM trace on stdout from this point in the file.
S	-	Optimize FastRAM for single-threaded simulations. Deprecated in Fast Models 11.28 and later because FastRAM automatically detects and optimises for a single-threaded context. If used, FastRAM outputs a warning and ignores it.
N	-	Disable MTE support with FastRAM if the platform has enabled MTE. This option is enabled by default. See <a href="#">FastRAM limitations</a> for requirements on the tag store.
F	-	Disable atomic memory operations through FastRAM. Enabled by default.
M	<string> or ALL	Identify the bus masters to use FastRAM, either masters whose id contains <code>string</code> or all masters. Can be specified multiple times.  <b>Note:</b> If the argument to <code>M</code> is not <code>ALL</code> and trace is enabled, then the ids of all masters are shown on the console with a message stating whether the master is enabled for FastRAM or not.  To find the list of masters, use <code>M foo</code> then use the list to select the masters required.
+	<base> <size>	Add physical address range from <base> to <base>+<size>
-	<base> <size>	Remove physical address range from <base> to <base>+<size>
=	<base-a> <base-b> <size>	Alias a physical address range
#	<text>	Comment

### Usage

The positions of `T` and `Q` trace options affect whether FastRAM trace covers initialization and/or runtime:

- To enable trace during the entire initialization and runtime, start the file with `T` and do not use `Q`.
- To enable trace during runtime only, end the file with `T` and do not use `Q`.
- To enable trace during initialization only, start the file with `T` and end the file with `Q`.
- To enable trace during specific parts of the initialization, use one or more pairs of `T` and `Q` within the file.

All addresses and sizes must be 64 MB-aligned (0x4000000) hexadecimal.

For more information, see:

- [FastRAM](#)
- [FastRAM limitations](#)

## 7.3 FastRAM configuration file example

This example FastRAM configuration file is written for a Base Platform FVP.

It does the following:

- Uses the `T` option at the start of the file to enable FastRAM trace output from the start of the FastRAM initialization.
- Enables FastRAM for the address range `0x08_00000000-0xff_ffffffff`.
- Defines `0x00_80000000-0x00_ffffffff` as an alias for the range `0x08_00000000-0x08_7fffffffff`.
- Uses the `Q` option at the end of the file to disable FastRAM trace output at the end of the FastRAM initialization.

```
# FastRAM config file for FVP Base
T
M ALL
+ 800000000 F800000000
= 80000000 800000000 80000000
Q
```

If FastRAM has been successfully enabled, it prints the following output:

```
FastRAM: CONSTRUCTED
FastRAM: Address space size = 40 bits
FastRAM: Slab size = 64 Mb
FastRAM: Page size = 4 kb
FastRAM: Singleton size = 147 kb
FastRAM: Number of monitors = 16
FastRAM: Enable ALL masters
FastRAM: Add range 0x08_00000000...ff_ffffffff
FastRAM: Add range 0x00_80000000...00_ffffffff
FastRAM: Alias range 0x00_80000000...00_ffffffff <=> 0x08_00000000...08_7fffffffff
```

For more information, see:

- [FastRAM configuration file syntax](#)
- [FastRAM](#)

## 7.4 FastRAM limitations

FastRAM can be used with most, but not all, platform models. This topic lists conditions where FastRAM is safe to use, and scenarios where it cannot be used, including MTE considerations.

It can be used in a platform in which all of the following conditions are true:

- The platform contains one or more very frequently accessed RAM components that are a whole multiple of 64 MB in size.
- These RAM components are always mapped to the same static physical range as seen by the bus masters that frequently access the RAM.
- The physical ranges used to access the RAM components by the bus masters can include aliased regions.
- The RAM components and the buses to them always give back DMI and for a given physical address always give back exactly the same DMI pointer and never invalidate DMI.
- Either all the bus masters in the platform use FastRAM for the configured physical ranges or you can identify the subset of masters that can use it by name. See [FastRAM configuration file syntax](#) for how to find the list of bus masters.
- All the bus masters that use FastRAM use the same physical address map to access the RAM components.
- If the RAM components internally allocate memory that is a whole multiple of 64 MB, then FastRAM can be used with RAM instances that are accessed by:
  - Bus masters that are enabled to use FastRAM.
  - Bus masters that are not, or cannot, be enabled to use FastRAM.
- If the RAM components internally allocate memory that is not a whole multiple of 64 MB, for example the RAMDevice LISA component, then FastRAM can only be used with RAM instances that are accessed by masters that are enabled to use FastRAM.

It cannot be used in a platform if any of the following conditions are true:

- Cache state modeling is enabled.
- The physical address map used by the bus masters to access the RAM is dynamic and can change at run time.
- The set of bus masters that will use FastRAM cannot be identified. See [FastRAM configuration file syntax](#) for how to find the list of bus masters.
- There is System IP between the bus masters and the RAM that needs to provide functionality other than a global monitor. However, a CCI or CCN with cache state modeling disabled is allowed.
- The platform RAM is mapped to an address greater than or equal to 0x100\_0000\_0000.
- The expected functionality of the platform depends on being able to invalidate DMI. FastRAM ignores DMI invalidations other than what is required internally to support exclusives and RevokeReadOnWrite behavior.

To enable MTE support through FastRAM:



- The platform must use a single system-wide tag store that returns DMI that can be safely extrapolated to 64 MB.
- If the platform uses a [CI700](#), or a similar model, where each SN-F has its own tag store, these must be disabled. See the `bypass_tag_cache` parameter on the model. Also, the platform must have a PVMetaDataController close to the DRAM.
- FastRAM does not support platforms that use tag carveout.

## 8. Fast Models examples

The following top-level example directories are installed under `$PVLIB_HOME/examples/`.

**Table 8-1: Fast Models examples directories**

Directory name	Description
LISA	LISA+ source code and project files for FVPs.
LISAPlus	Example LISA+ components that show how to capture and generate MTI trace, remap PVBUS transactions, and handle burst transactions.
MTI	MTI plug-in examples that show how to extract and use trace information from models. These examples are also available as pre-built libraries under <code>\$PVLIB_HOME/plugins/</code> : <ul style="list-style-type: none"> <li>GenericCounter</li> <li>GenericTrace</li> <li>ListTraceSources</li> </ul>
SystemCExport	<ul style="list-style-type: none"> <li>Source code and makefiles for EVS platform examples and SVPs</li> <li>LISA+ source for bridges and EVS components</li> <li>Header files required for exporting LISA+ protocols to SystemC</li> </ul>

### 8.1 LISA examples

LISA+ source and project files for FVPs.



Note

The LISA platform examples are Integrated SIMulators (ISIMs). For more information about building and running them, see [Build and run an FVP example](#).

The following LISA examples are provided:

**Table 8-2: LISA examples**

Example	Description
BusComponents	Example LISA+ components that demonstrate different ways of using the PVBUS interface.
Common	FVP-specific LISA+ components that are common to different types of FVPs.
CSS	Source and project files for Reference Design FVPs. These FVPs model compute subsystems (CSS) that target specific market segments. Reference software stacks are available for them, see <a href="#">Arm Ecosystem FVPs</a> for more information.
FVP_Base	Source and project files for Base Platform FVP examples. For information about the Base Platform, see <a href="#">Base Platform</a> .
FVP_BaseR	Source and project files for BaseR Platform FVP examples.
FVP_Base_RevC	Source and project files for Base Platform RevC FVP examples. For information about the Base Platform RevC, see <a href="#">Base Platform RevC</a> .
FVP_Coproc_Demo	Example implementation of the Coprocessor interface. Registers the coprocessor with a ARMCortexM33CT or ARMAEMv8MCT model. For more information, see <a href="#">CoprocBusProtocol protocol</a> .

Example	Description
FVP_MPS2	Source and project files for MPS2-based example platforms. For information about the MPS2 platforms, see <a href="#">MPS2</a> .
FVP_MPS3	Source and project files for MPS3-based example platforms that support the Arm®Corstone™ SSE-300 Example Subsystem. For more information, see <a href="#">Arm Corstone SSE-300 Example Subsystem Technical Reference Manual</a> .
FVP_VE	Source and project files for VE FVPs. For information about the VE platform, see <a href="#">Versatile Express</a> .
VP_PChannel	Shows how to create power controllers to control the power state of the cores and cluster, using the PChannel protocol. For information about PChannel, see <a href="#">PChannel protocol</a> .

## 8.2 Build and run an FVP example

Follow these instructions to build and run a Base Platform FVP example. The FVP examples are located under `$PVLIB_HOME/examples/LISA/`.

### Before you begin

Ensure the following:

- You have installed Fast Models and SystemC 2.3.4 and have set the environment variables as described in [Installing Fast Models](#) in the Fast Models User Guide.
- You have set up a Fast Models license, either using FlexNet Licensing or [User-Based Licensing](#).
- You are using a supported Operating System and have set up a compatible toolchain, from those listed in [Requirements for Fast Models](#) in the Fast Models User Guide.

### About this task

The build uses the `simgen` command from a terminal to output an executable platform model called an ISIM (Integrated SIMulator).

### Procedure

1. Use the following commands to build an FVP example:

```
cd $PVLIB_HOME/examples/LISA/FVP_Base/Build_Cortex-A55
simgen -b -p FVP_Base_Cortex-A55.sgproj --configuration Linux64-Release-GCC-9.3
```

where:

**-b**

means build the target.

**-p**

specifies the SimGen project file. It has a `.sgproj` extension.

**--configuration**

specifies the build configuration name. In this example, we are performing a release build on Linux using GCC 9.3.

This command generates an executable named `isim_system` in `$PVLIB_HOME/examples/LISA/FVP_Base/Build_Cortex-A55/Linux64-Release-GCC-9.3/`.

2. To run `isim_system` from the terminal:

- a. Navigate to the directory where it is located.
- b. To see a full list of command-line options for `isim_system`, run it with the `--help` option:

```
./isim_system --help
```

- c. The following example command line shows how to load an application on `isim_system`:

```
./isim_system -a FVP_Base_Cortex_A55.cluster0.*=$PVLIB_HOME/images/image.axf -  
C bp.secure_memory=0
```

where:

**-a**

is the name of the application to load. Optionally it also specifies which core instances to run it on, in this case all cores in cluster0.

**-c**

is a configuration parameter. To see a full list of the available parameters, run the model with the `--list-params` option. To specify multiple parameters, it might be more convenient to place them in a text file, each parameter on a new line, and pass them to the model when starting it using `--config-file <filename>`.

- d. The ISIM starts running, displaying the output in a CLCD window.
- e. To exit the simulation, press **Ctrl+C**.

### Related information

[System Generator](#)

[Fast Models glossary](#)

[LISA examples](#) on page 6478

## 8.3 LISAPlus examples

Source code and SimGen project files for example LISA+ components.

**Table 8-3: LISAPlus examples**

Example	Description
CapturingTraceFromLISA	Instantiates and uses an MTI plug-in from a LISA+ component. Uses the SimpleTrace plug-in as an example.
GeneratingTraceFromLISA	Generates MTI trace information from a LISA+ component.
PVBusBursts	Uses a <a href="#">PVBusMaster</a> to generate burst read transactions, which are handled by a <a href="#">PVBusSlave</a> .
RemappingWithPVBusMapper	Uses the <a href="#">PVBusMapper</a> component to remap transactions based on their attributes.

## 8.4 MTI examples

Example MTI plug-ins that show how to use MTI to extract and use trace information from models.

The following MTI examples are provided:

**Table 8-4: MTI examples**

Example	Description
CallTrace	Displays a function call sequence by tracing the PC field of INST trace sources, then compares the output with values in a symbol table. See the <a href="#">readme</a> for more information.
CountingCacheStats	Registers counters for cache-related trace sources, for example CACHE_READ_HIT. Prints the cache stats before terminating.
DCCTrace	Prints the value of DBGDTRxx_EL0 when data is written. Updates the TxFull bit in MDSCR to indicate the data was read. See the <a href="#">readme</a> for more information.
GenericCounter	Registers a counter for trace sources. Prints the counter value for each INST trace source before terminating.  This example is also available as a pre-built library, see <a href="#">GenericCounter</a> .
GenericTrace	A flexible plug-in that traces one or more trace sources specified by the user. Prints the trace to a text file or to stdout.  This example is also available as a pre-built library, see <a href="#">GenericTrace</a> .
ITMtrace	Captures instrumentation trace macrocell (ITM) packets, which enables you to use ITM with a Cortex®-M class model.  For more information about this plug-in, see <a href="#">Trace Cortex-M software with the Instrumentation Trace Macrocell (ITM)</a> on Arm Community.
ListTraceSources	Displays either the trace sources provided by all trace components in the model, or just the trace components, to a text file or to stdout, without running the simulation. For more information, see <a href="#">readme.txt</a> .  This example is also available as a pre-built library, see <a href="#">ListTraceSources</a> .
RunTimeParameterTest	Uses MTI to set runtime parameters.
SimpleTrace	Simple trace plug-in that prints a trace of the PC.
SoftwareTrigger	Traces SEMIHOSTING_PRECALL trace events, intercepts semihosting calls, and prints out register information. For more information, see the <a href="#">readme</a> .
TraceOnBreak	Similar to the SimpleTrace example, but prints the PC value only when a breakpoint is hit.

### Related information

[Fast Models Model Trace Interface Reference Manual](#)

## 8.5 SystemCExport examples

Components and platform models that are created by exporting LISA+ components or platforms to SystemC. Also, bridge components for converting transactions between LISA+ protocols and SystemC.

**Table 8-5: SystemCExport examples**

Directory	Description
Bridges	LISA+ source for bridge components.
Common	Source files and makefile rules that are common to the EVS and SVP examples.
Common/ Protocols	Header files that are required for the export of LISA+ protocols to SystemC.
EVS_Components	LISA+ files and project files for EVS (Exported Virtual Subsystem) components. These are LISA+ components with a SystemC wrapper and bridges that allow them to be used in a SystemC simulation.
EVS_Platforms	<p>LISA+ source and makefiles for EVS platform examples.</p> <p>An EVS platform is a LISA+ platform that has been exported as a SystemC object to allow it to be integrated into a SystemC simulation.</p> <p>The EVS platform examples are minimal platforms that are designed for a specific use case, for example running the Dhrystone benchmark application or booting Linux.</p> <p>The Fast Models package installs the Dhrystone image <code>dhrystone_v8.axf</code> in the <code>\$PVLIB_HOME/images/</code> directory.</p> <p>For more information about building an EVS platform, see <a href="#">Build and run an EVS platform example</a>.</p>
SVP_Platforms	<p>SVPs (SystemC Virtual Platforms) are platform models in which each component or subsystem has been individually exported to SystemC using the Fast Models Multiple Instantiation (MI) feature. For more information, see <a href="#">SVP build target</a> in the Fast Models User Guide.</p> <p>SVP platforms can be modified by replacing EVS components with other Fast Models EVSs, or with native SystemC components.</p> <p>For more information about building an SVP, see <a href="#">Build and run an SVP example</a>.</p>

## 8.6 Build and run an EVS platform example

The EVS platform examples are located under `$PVLIB_HOME/examples/SystemCExport/EVS_Platforms/`.

### Before you begin

Ensure the following:

- You have installed Fast Models and SystemC 2.3.4 and have set the environment variables as described in [Installing Fast Models](#) in the Fast Models User Guide.
- You have set up a Fast Models license, either using FlexNet Licensing or [User-Based Licensing](#).
- You are using a supported Operating System and have set up a compatible toolchain, from those listed in [Requirements for Fast Models](#) in the Fast Models User Guide.

## About this task

Follow these instructions to build and run one of the EVS platform examples. This tutorial uses the EVS\_Dhrystone\_Cortex-A75x1 example, which is a minimal platform that is designed to run the Dhrystone benchmark application.

## Procedure

1. These examples are built using a Makefile. Open a terminal and navigate to the directory containing the example, and run `make`, specifying the build configuration, for example:

```
cd $PVLIB_HOME/examples/SystemCExport/EVS_Platforms/EVS_Dhrystone/Build_Cortex-A75x1
make rel_gcc93_64
```

This command creates the target executable `evs_dhrystone_cortex-a75x1.x` and copies into the current directory the shared object files that are required to run it.

2. To see a full list of command-line options for the EVS platform, run it with the `--help` option:

```
./EVS_Dhrystone_Cortex-A75x1.x --help
```

3. From the terminal, enter the following command to launch the platform and load the Dhrystone image, `dhrystone_v8.axf`, specifying 10000000 as the number of runs through the benchmark:

```
echo 10000000 | ./EVS_Dhrystone_Cortex-A75x1.x -a $PVLIB_HOME/images/dhrystone_v8.axf
```

4. The simulation runs, printing output to the terminal.

---

When loading an image on an EVS, you might see the following warning:



Note

```
Warning: Base.cluster0.cpu0: Uncaught exception, thread terminated
In file: gen/scx_scheduler_mapping.cpp:523
In process: Base.thread_p_5 @ 0 s
```

This warning means that the image is attempting to run from DRAM, but this is access-controlled by the TZC\_400 component. To disable security checking by the TZC\_400, specify `-c Base.bp.secure_memory=false` when running the EVS.

---

## 8.7 Build and run an SVP example

The SystemC Virtual Platform (SVP) examples are located under `$PVLIB_HOME/examples/SystemCExport/SVP_Platforms/`.

### Before you begin

Ensure the following:

- You have installed Fast Models and SystemC 2.3.4 and have set the environment variables as described in [Installing Fast Models](#) in the Fast Models User Guide.
- You have set up a Fast Models license, either using FlexNet Licensing or [User-Based Licensing](#).

- You are using a supported Operating System and have set up a compatible toolchain from those listed in [Requirements for Fast Models](#) in the Fast Models User Guide.

## About this task

Follow these instructions to build and run one of the SVP examples.

## Procedure

1. These examples are built using a Makefile. Open a terminal, navigate to the directory containing the example, and run `make`, specifying the build configuration, for example:

```
cd $PVLIB_HOME/examples/SystemCEExport/SVP_Platforms/SVP_Base/Build_Cortex-A57x1
make rel_gcc93_64
```

This command creates the target executable `svp_Base_Cortex-A57x1.x` and copies into the current directory the shared object files that are required to run the platform.

2. To see a full list of command-line options for the platform, run it with the `--help` option:

```
./SVP_Base_Cortex-A57x1.x --help
```

3. Optionally, some example images are available in the Third Party Add-ons for Fast Models package. Download this package from [Product Download Hub](#).
4. Install the package to the location of your existing Fast Models installation. The package installs the images into `$PVLIB_HOME/images/`.
5. The following example command-line launches the platform, using the `-a` option to load one of the images:

```
./SVP_Base_Cortex-A57x1.x -a $PVLIB_HOME/images/brot_ve_64.axf
```

6. The simulation starts running, displaying the output in a CLCD window.
7. To exit the simulation, press **Ctrl+C**.



# Proprietary Notice

This document is protected by copyright and other related rights and the use or implementation of the information contained in this document may be protected by one or more patents or pending patent applications. No part of this document may be reproduced in any form by any means without the express prior written permission of Arm Limited ("Arm"). No license, express or implied, by estoppel or otherwise to any intellectual property rights is granted by this document unless specifically stated.

Your access to the information in this document is conditional upon your acceptance that you will not use or permit others to use the information for the purposes of determining whether the subject matter of this document infringes any third party patents.

The content of this document is informational only. Any solutions presented herein are subject to changing conditions, information, scope, and data. This document was produced using reasonable efforts based on information available as of the date of issue of this document. The scope of information in this document may exceed that which Arm is required to provide, and such additional information is merely intended to further assist the recipient and does not represent Arm's view of the scope of its obligations. You acknowledge and agree that you possess the necessary expertise in system security and functional safety and that you shall be solely responsible for compliance with all legal, regulatory, safety and security related requirements concerning your products, notwithstanding any information or support that may be provided by Arm herein. In addition, you are responsible for any applications which are used in conjunction with any Arm technology described in this document, and to minimize risks, adequate design and operating safeguards should be provided for by you.

This document may include technical inaccuracies or typographical errors. THIS DOCUMENT IS PROVIDED "AS IS". ARM PROVIDES NO REPRESENTATIONS AND NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, SATISFACTORY QUALITY, NON-INFRINGEMENT OR FITNESS FOR A PARTICULAR PURPOSE WITH RESPECT TO THE DOCUMENT. For the avoidance of doubt, Arm makes no representation with respect to, and has undertaken no analysis to identify or understand the scope and content of, any patents, copyrights, trade secrets, trademarks, or other rights.

TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL ARM BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF ARM HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Reference by Arm to any third party's products or services within this document is not an express or implied approval or endorsement of the use thereof.

This document consists solely of commercial items. You shall be responsible for ensuring that any permitted use, duplication, or disclosure of this document complies fully with any relevant

export laws and regulations to assure that this document or any portion thereof is not exported, directly or indirectly, in violation of such export laws. Use of the word “partner” in reference to Arm’s customers is not intended to create or refer to any partnership relationship with any other company. Arm may make changes to this document at any time and without notice.

This document may be translated into other languages for convenience, and you agree that if there is any conflict between the English version of this document and any translation, the terms of the English version of this document shall prevail.

The validity, construction and performance of this notice shall be governed by English Law.

The Arm corporate logo and words marked with ® or ™ are registered trademarks or trademarks of Arm Limited (or its affiliates) in the US and/or elsewhere. Please follow Arm’s trademark usage guidelines at <https://www.arm.com/company/policies/trademarks>. All rights reserved. Other brands and names mentioned in this document may be the trademarks of their respective owners.

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

PRE-1121-V1.0

# Product and document information

Read the information in these sections to understand the release status of the product and documentation, and the conventions used in Arm documents.

## Product status

All products and services provided by Arm require deliverables to be prepared and made available at different levels of completeness. The information in this document indicates the appropriate level of completeness for the associated deliverables.

### Product completeness status

The information in this document is Final, that is for a developed product.

## Revision history

These sections can help you understand how the document has changed over time.

### Document release information

The Document history table gives the issue number and the released date for each released issue of this document.

#### Document history

Issue	Date	Confidentiality	Change
1131-00	4 March 2026	Non-Confidential	Update for v11.31
1130-00	19 November 2025	Non-Confidential	Update for v11.30
1129-00	16 May 2025	Non-Confidential	Update for v11.29
1128-00	19 February 2025	Non-Confidential	Update for v11.28
1127-00	16 September 2024	Non-Confidential	Update for v11.27
1126-00	19 June 2024	Non-Confidential	Update for v11.26

Issue	Date	Confidentiality	Change
1125-00	13 March 2024	Non-Confidential	Update for v11.25.
1124-00	6 December 2023	Non-Confidential	Update for v11.24.
1123-00	13 September 2023	Non-Confidential	Update for v11.23.
1122-00	14 June 2023	Non-Confidential	Update for v11.22.
1121-00	22 March 2023	Non-Confidential	Update for v11.21.
1120-00	7 December 2022	Non-Confidential	Update for v11.20.
1119-00	14 September 2022	Non-Confidential	Update for v11.19.
1118-00	15 June 2022	Non-Confidential	Update for v11.18.
1117-00	16 February 2022	Non-Confidential	Update for v11.17.
1116-00	6 October 2021	Non-Confidential	Update for v11.16.
1115-00	29 June 2021	Non-Confidential	Update for v11.15.
1114-01	30 March 2021	Non-Confidential	Document update 1 for v11.14.
1114-00	17 March 2021	Non-Confidential	Update for v11.14.
1113-00	9 December 2020	Non-Confidential	Update for v11.13.
1112-00	22 September 2020	Non-Confidential	Update for v11.12.
1111-00	9 June 2020	Non-Confidential	Update for v11.11.
1110-00	12 March 2020	Non-Confidential	Update for v11.10.

Issue	Date	Confidentiality	Change
1109-00	28 November 2019	Non-Confidential	Update for v11.9.
1108-00	5 September 2019	Non-Confidential	Update for v11.8.
1107-00	17 May 2019	Non-Confidential	Update for v11.7.
1106-01	25 March 2019	Non-Confidential	Update for v11.6.1.
1106-00	27 February 2019	Non-Confidential	Update for v11.6.
1105-00	23 November 2018	Non-Confidential	Update for v11.5.
1104-01	17 August 2018	Non-Confidential	Update for v11.4.2.
1104-00	22 June 2018	Non-Confidential	Update for v11.4.
1103-00	23 February 2018	Non-Confidential	Update for v11.3.
1102-00	17 November 2017	Non-Confidential	Update for v11.2.
1101-00	31 August 2017	Non-Confidential	Update for v11.1.
1100-00	31 May 2017	Non-Confidential	Update for v11.0. Document numbering scheme has changed.

## Change history

For information about the functional changes to Fast Models, see the [Fast Models Release Notes](#).

## Conventions

The following subsections describe conventions used in Arm documents.

### Glossary


The Arm Glossary is a list of terms used in Arm documentation, together with definitions for those terms. The Arm Glossary does not contain terms that are industry standard unless the Arm meaning differs from the generally accepted meaning.

See the Arm Glossary for more information: [developer.arm.com/glossary](https://developer.arm.com/glossary).

Typographic conventions


Arm documentation uses typographical conventions to convey specific meaning.

Convention	Use
italic	Citations.
<b>bold</b>	Interface elements, such as menu names.  Terms in descriptive lists, where appropriate.
monospace	Text that you can enter at the keyboard, such as commands, file and program names, and source code.
monospace <u>underline</u>	A permitted abbreviation for a command or option. You can enter the underlined text instead of the full command or option name.
<and>	Encloses replaceable terms for assembler syntax where they appear in code or code fragments.  For example: <div>MRC p15, 0, &lt;Rd&gt;, &lt;CRn&gt;, &lt;CRm&gt;, &lt;Opcode_2&gt;</div>
SMALL CAPITALS	Terms that have specific technical meanings as defined in the <i>Arm® Glossary</i> . For example, <b>IMPLEMENTATION DEFINED</b> , <b>IMPLEMENTATION SPECIFIC</b> , <b>UNKNOWN</b> , and <b>UNPREDICTABLE</b> .




Caution

We recommend the following. If you do not follow these recommendations your system might not work.




Warning

Your system requires the following. If you do not follow these requirements your system will not work.



Danger

You are at risk of causing permanent damage to your system or your equipment, or harming yourself.



Note

This information is important and needs your attention.



A useful tip that might make it easier, better or faster to perform a task.

---



**Remember**

A reminder of something important that relates to the information you are reading.

---

# Useful resources

This document contains information that is specific to this product. See the following resources for other useful information.

Arm documents are available on [developer.arm.com/documentation](https://developer.arm.com/documentation).

Confidential documents are only available to licensees, when logged in. Each document link in the following tables provides direct access to the online version of the document.

Arm product resources	Document ID	Confidentiality
<a href="#">A-Profile Architecture</a>	–	Non-Confidential
<a href="#">AMBA-PV Extensions to TLM User Guide</a>	100962	Non-Confidential
<a href="#">AN521 Example SSE-200 Subsystem for MPS2+ Application Note</a>	DAI 0521	Non-Confidential
<a href="#">Arm® Architecture Models</a>	–	Non-Confidential
<a href="#">Arm® Corstone™ SSE-300 Example Subsystem Technical Reference Manual</a>	101773	Non-Confidential
<a href="#">Arm® Cortex®-M23 Armv8-M IoT Kit User Guide</a>	ECM 0635473	Non-Confidential
<a href="#">Arm® Cortex®-M33 processor Armv8-M IoT Kit FVP User Guide</a>	ECM 0601256	Non-Confidential
<a href="#">Arm® Cortex®-M7 SMM on V2M-MPS2 Application Note 400</a>	DAI 0400	Non-Confidential
<a href="#">Arm® Development Studio Getting Started Guide</a>	101469	Non-Confidential
<a href="#">Arm® DynamIQ Shared Unit Technical Reference Manual</a>	100453	Non-Confidential
<a href="#">Arm® Socrates™</a>	–	Non-Confidential
<a href="#">Fast Models FVPs in Arm Development Studio Reference Guide</a>	110379	Non-Confidential
<a href="#">Fast Models Model Trace Interface Reference Manual</a>	DUI 0819	Non-Confidential
<a href="#">Fast Models Tools User Guide</a>	109415	Non-Confidential
<a href="#">Fast Models User Guide</a>	100965	Non-Confidential
<a href="#">Fixed Virtual Platforms</a>	–	Non-Confidential
<a href="#">How to generate ASTF traces of workloads running on Fast Models</a>	109193	Non-Confidential
<a href="#">Introduction to SVE2</a>	102340	Non-Confidential
<a href="#">LISA+ Language for Fast Models Reference Guide</a>	101092	Non-Confidential
<a href="#">Motherboard Express µATX V2M-P1 Technical Reference Manual</a>	DUI 0447	Non-Confidential
<a href="#">Product Download Hub</a>	–	Non-Confidential
<a href="#">Tarmac Trace Utilities</a>	–	Non-Confidential
<a href="#">Workload Trace Generation Best Practices</a>	107983	Non-Confidential

Arm architecture and specifications	Document ID	Confidentiality
<a href="#">AMBA® Low Power Interface Specification</a>	IHI 0068	Non-Confidential
<a href="#">Armv7-M Architecture Reference Manual</a>	DDI 0403	Non-Confidential
<a href="#">Arm® Architecture Reference Manual Supplement, The Scalable Vector Extension</a>	DDI 0584	Non-Confidential
<a href="#">Arm® Architecture Reference Manual for A-profile architecture</a>	DDI 0487	Non-Confidential
<a href="#">Arm® Generic Interrupt Controller Architecture version 2.0 - Architecture Specification</a>	IHI 0048	Non-Confidential



Arm architecture and specifications	Document ID	Confidentiality
<a href="#">Arm® Power Policy Unit Architecture Specification</a>	DEN 0051	Non-Confidential
<a href="#">Arm® Realm Management Extension (RME) System Architecture</a>	DEN 0129	Non-Confidential

Non-Arm resources	Document ID	Organization
<a href="#">Accellera Systems Initiative</a>	–	<a href="https://www.accellera.org">https://www.accellera.org</a>
<a href="#">GitHub</a>	–	Github
<a href="#">Intel Download Center, Intel StrataFlash Memory (J3) datasheet</a>	–	Intel
<a href="#">MultiMedia Card Association specification</a>	–	JEDEC
<a href="#">Simple DirectMedia Layer Cross-platform Development Library</a>	–	Simple DirectMedia Layer
<a href="#">Virtual I/O Device (VIRTIO) Version 1.0</a>	–	OASIS Open